

MICROBITS

MACROINFORMAÇÕES E PROGRAMAS PARA MICROS DA LINHA SINCLAIR (TK 82/83/85, TK 90X, CP 200 E COMPATÍVEIS)

DQ

**Simulando
um Computador**

**Aumente
o Vocabulário**

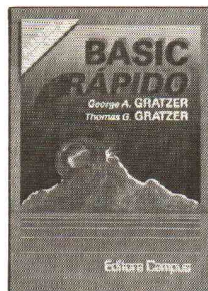
**Jogos
dos Buracos**



RACECOLOR
Para TK 90X

**BASIC RÁPIDO:
ALÉM DO BASIC TRS-80**
G. A. Gratzler e T. G. Gratzler
1985 - 200 pp.

Qualquer coisa que você faça com o BASIC do seu TRS-80, BASIC RÁPIDO faz ainda melhor! Enfim a velocidade e eficiência que você desejava nas respostas de seu CP 500, DGT-1000, SYSDATA III e compatíveis.



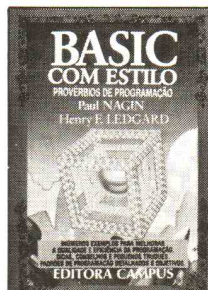
PROGRAMAÇÃO EM BASIC
John G. Kemeny & Thomas E. Kurtz
1985 - 258 pp.

O objetivo deste texto é o de oferecer uma introdução simples à programação de computadores utilizando a linguagem BASIC, passando, então, a descrever uma ampla variedade de aplicações de interesse prático.



ENCICLOPÉDIA DE LINGUAGEM BASIC
Clóvis Pereira & Rossana Benvenuti Alcântara
1985 - 506 pp.

Apresenta ao usuário os comandos, instruções e funções nos equipamentos das diversas linhas, esclarecendo-lhe as dúvidas e fornecendo-lhe os elementos indispensáveis para o máximo aproveitamento dos mesmos.



**BASIC COM ESTILO:
PROVÉRBIOS DE PROGRAMAÇÃO**
Paul Nagin & Henry F. Ledgard
1985 - 144 pp.

Uma coleção de 19 "provérbios" - diretrizes e regras simples que visam tornar o programador de BASIC capaz de escrever programas elegantes, legíveis e isentos de erros, abordando ainda programação Top-Down e alguns tópicos atuais e controversos.

BASES PARA O SEU BASIC



BASIC BÁSICO - 5. Edição
Jorge da Cunha Pereira Filho
1985 - 248 pp.
"Bíblia do Basic no Brasil", indispensável para quem deseja conhecer esta imprescindível linguagem. Os aspectos mais fundamentais são abordados nas 5 "lições", complementadas por exercícios práticos e por um resumo para consulta e referência.



BASIC PARA MICROS PESSOAIS
2. Edição
Jorge da Cunha Pereira Filho
1984 - 232 pp.
Do mesmo autor de Basic Básico, a "Bíblia do Basic no Brasil", fornece ao leitor as informações necessárias para a perfeita utilização desta linguagem nos microcomputadores pessoais.

BASIC SINCLAIR
Raul Udo Christmann
1985 - 124 pp.

Primeira obra sobre Linguagem BASIC especificamente destinada aos possuidores e usuários da popular Linha Sinclair, inclui numerosos exemplos, dicas e problemas propostos, sendo os assuntos abordados em ordem crescente de dificuldade.



BASIC PARA APLICAÇÕES COMERCIAIS
Douglas Hergert
1984 - 208 pp.

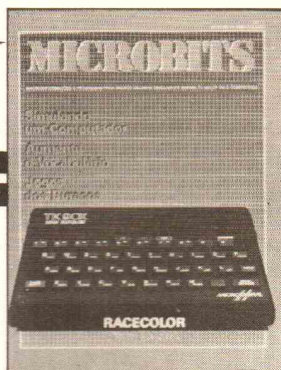
A melhor maneira de descobrir o que um computador pode fazer é aprendendo a programar. Esta é uma introdução ao BASIC destinada a ensinar a escrever, ler e depurar programas na mais popular das linguagens de programação.



Editora Campus Ltda.

LIVROS CIENTÍFICOS E TÉCNICOS

Rua Barão de Itapagipe, 55 - 20261 - Rio de Janeiro - RJ - Tel.: (021) - 284-8443



Editorial

Diretor Editorial
David Thomas Anderson

Editor Responsável
Sebastião Miranda

Consultor Técnico
Bernhard Wolfgang Schön

Redação
Eni Barreto Anderson

Programação Visual
Otavio Studart

Arte
Valmir Oliveira

Colaboradores
Nelson N. S. Santos, Alexandre Gromow, Alexandre Pfeifer, Bernard Wolfgang Schon, Daniel Quadros, Rogerio Colonna dos Santos, Sergio Augusto Moraes, Valter Pereira Viana e Waldir Costa Sola.

Publicidade
Rio de Janeiro
Editora Campus Ltda.
Rua Barão de Itapagipe 55 Rio Comprido
CEP 20261 RJ - RJ - Tel.: (021) 284-8443
São Paulo

MICRO BITS Informática Ltda.
Cx. Postal 12464 - CEP 04798 SP - SP
Tel.: (011) 521 4235

Circulação e Assinaturas
Editora Campus Ltda.

Distribuição
Fernando Chinaglia
Rua Teodoro da Silva, 907 - RJ Tel.: (021) 268 9112
Impressão e Acabamento

MICROBITS é uma publicação bimestral da Editora Campus Ltda., Rua Barão de Itapagipe 55 Rio Comprido - RJ CEP 20261 Tel.: (021) 284-8443

© 1985, Editora Campus Ltda.

Todos os direitos reservados e protegidos pela Lei 5988 de 14/12/1973.

Nenhuma parte desta revista poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação, ou quaisquer outros.

Todo o esforço foi feito para fornecer a mais completa e adequada informação. Contudo a editora não assume responsabilidades pelo uso da mesma.

A Editora Campus não é filiada a nenhum fabricante de sistemas computacionais.

CORRESPONDÊNCIA

1. Assinaturas

Desejando obter informações sobre assinaturas, escreva para:

"Assinaturas"

Revista MICROBITS

A/c Editora Campus Ltda.

Rua Barão de Itapagipe 55 - 20261 Rio de Janeiro, RJ
Tel.: (021) 284-8443

2. Redação

Envie suas cartas, programas e artigos para:

Ao Diretor Editorial - Revista MICROBITS

Caixa Postal 12464 - 04798 São Paulo, SP.

LEMBRETE IMPORTANTE

Esclarecemos mais uma vez aos nossos leitores que a Editora Campus não é responsável por quaisquer questões relativas aos números de MICROBITS anteriores ao Ano II - nº 1. Assim, solicitamos que as mesmas sejam diretamente tratadas com o Diretor Editorial, Sr. David Anderson, no seguinte endereço: Caixa Postal 12464 - 04798 - São Paulo - SP

O lançamento do TK90X pela Microdigital trouxe nova vida ao mercado de micros pessoais no Brasil, há muito tempo em estagnação pela falta de novidades. Embora o número de aficcionados dos micros TK82/83/85, CP-200 e similares continue grande (e deva continuar assim por bastante tempo), há muitos usuários sentindo falta de equipamentos mais sofisticados, com cor, som, alta resolução e outros recursos, num micro de fácil acesso. Este novo modelo, baseado no micro de maior sucesso mundial - O ZX Spectrum - dispõe da maioria dos recursos desejados e, se as previsões da Microdigital se realizarem, haverá 200.000 destas máquinas espalhadas pelo Brasil até o final de 1986.

Mas não é lógico esperar que outros fabricantes tradicionais (e os que ainda não entraram na "corrida") deixem de concorrer com o TK90X. Há rumores de que firmas como Gradient e Sharp entrarão nesta faixa do mercado. E quem sabe a Prológica não tem um ás escondido na manga?

Um micro que aparece muito nestes boatos é o modelo MSX, baseado no sistema da MICROSOFT, que já tem muitas firmas (principalmente japonesas - SONY, SANYO, TOSHIBA, JVC etc.) com suas máquinas à venda no mundo inteiro. (Menos no Brasil, é claro.) Embora com bons recursos, ainda não conseguiu vender bem na Europa e muito menos nos Estados Unidos, onde o mercado para micros pequenos não é dos melhores. Também a falta de software, comparado ao que há disponível para os Apple's, TRS-80's e ZX-Spectrum's é notável. Afinal, um micro sem software nunca será um sucesso.

Um fator pode mudar muito este relativamente fraco desempenho mundial, colocando o MSX na posição número 1 em vendas. Este fator se chama IBM. A IBM, cujo único micro pessoal, o PC Jr, foi um fracasso total (e parou de ser fabricado), está pesquisando a possibilidade de produzir uma versão MSX por preço atraente. Se for considerado um mercado rentável, talvez vejamos o mundo inteiro sob o domínio dos computadores IBM - desde os grandes, até os micros profissionais e pessoais.

Aguardemos as novidades - especialmente aqui no Brasil!

David Thomas Anderson

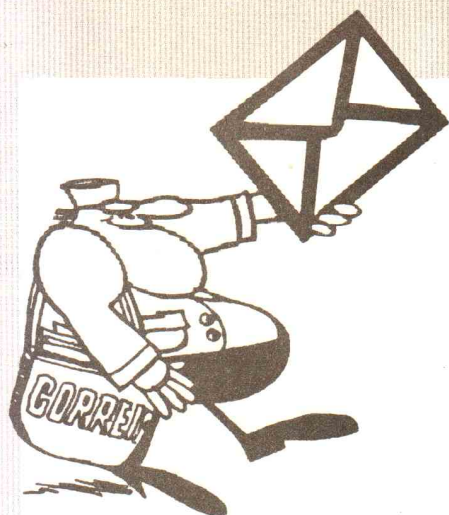
A Editora CAMPUS e MICROBITS, mantendo a posição de vanguarda, vão acompanhar o TK90X, lançando o que de melhor houver no mercado nacional e internacional para este fantástico equipamento, em termos de livros, programas e artigos.

Bem-vindo, TK90X.

Editora CAMPUS

Índice

Cartas	4
Desafio/Charada	5
Racecolor para TK90X	6
Traço	8
Simulando um computador	10
Sem o uso do REM	15
Jogo dos buracos	16
Além do BASIC	18
Como aumentar o vocabulário	22
Notícias	24
Linha Vertical em LM	26
Classificados/Dicas	29
Livros	30



Cartas

● Sou assinante da Revista MICROBITS que é para mim a revista de realment grande precisão.

Venho através desta solicitar a vocês uma dica de como fazer uma linha vertical no TK 85, com qualquer caractere, em qualquer coluna, em linguagem de máquina (Assembly).

Se não for muita importunação, que me informassem como escrever uma frase, também em linguagem de máquina.

JOÃO BATISTA NEGRO
Maringá – PR

μBits: Para não ocupar toda esta coluna com a resposta à sua pergunta, João, foi escrito o artigo "Linha Vertical em Linguagem de Máquina", publicado nesta edição na página 26.

● Possuindo um micro da lógica Sinclair (TK-85) há cerca de 3 meses, venho, como seria lógico, procurando revistas especializadas no ramo. Foi quando me informaram da existência de MICROBITS, que procurei exaustivamente, não a encontrando.

Eis que ontem, por um acaso, encontrei o número 3, ano II e fiquei gratamente surpreso ao ver a excelente revista, atendendo a todos os meus anseios naquilo que estava procurando.

Devo parabenizá-los pelo magnífico trabalho, que penso não ter paralelo nas congêneres existentes. Apesar de serem também de boa qualidade, não satisfazem plenamente aos usuários da linha Sinclair.

HUGO DUARTE SILVA KRESCH
São Bernardo do Campo – SP

μBits: Esperamos que continue a ler a nossa revista, Hugo, e que nos envie seus comentários, de vez em quando, sobre os programas e artigos publicados.

● Achei o nº 3, ano II da revista MICROBITS o maior barato! Considerei uma ótima idéia a nova seção "Desafio". Infelizmente, não consegui "matar" a charada.

Gostaria que vocês, se possível, falassem de um periférico desconhecido, mas que é importante para um jogo animado ficar muito mais animado: o Gerador de Som. Não é uma boa sugestão?

MARLOS MEDEIROS
Fortaleza – CE

μBits: Você, como muitos outros, Marlos, tem solicitado informações sobre um gerador de som. Esperamos analisar este equipamento para publicação numa outra edição.

● Quando adquiri o primeiro exemplar da MICROBITS, não esperava, sinceramente, que o nível editorial e técnico da revista chegasse a tanto. Devo lhe afirmar que cheguei a ficar meio "desconfiado", pois nunca tinha visto esta revista no distribuidor que costumo adquirir minhas publicações – arisquei e não me arrependo! E já que no editorial você faz um convite aos leitores para que lhes escrevam enviando sugestões, programas, etc., aqui estou eu. Hoje não estou enviando nada, pois no momento não tenho material que possa servir para a revista MICROBITS, mas vale para um contato inicial.

Adquiri, recentemente, um CP200-S da Prológica e gostaria, se possível, de saber se já existem no mercado os tais joysticks e as expansões de memória para esse computador, pois nas lojas que tenho consultado todos desconhecem os equipamentos mencionados. Também gostaria de saber qual a impressora que eu poderia ligar ao CP 200-S (isto se for possível, é claro) e onde posso adquirir uma. Por que preço?

ADALBERTO SCARDELAI
São Paulo – SP

μBits: Agradecemos os seus comentários sobre a nossa revista, Adalberto. Houve, realmente, alguma dificuldade na distribuição inicial de MICROBITS,

mas parece ter sido sanada completamente.

Quanto a sua pergunta sobre o CP 200-S, nós também não conhecemos nenhum periférico que possa ser utilizado com este micro. Isto, entretanto, não significa que não haja nada planejado pela Prológica ou outras firmas para suprir esta deficiência, embora a possibilidade pareça remota. Como a placa do CP-200 (e depois o CP 200-S) foi totalmente reprojetaada, tendo pouca coisa a ver com o original ZX-81, a placa de expansão não é compatível com nenhum dos periféricos que podem ser utilizados com a maioria dos micros TK82-C, TK83, TK85 e TK90X. Caso algum dos nossos leitores conheça algum produto que sirva para o CP 200-S, gostaríamos de receber maiores detalhes para divulgação na próxima edição.

● Tenho os dois primeiros números da MICROBITS, e gostei muito dos artigos que ela traz, principalmente o assunto que fala sobre "Caracteres Especiais", na parte II, o programa que fizeram está pra lá de bom. Mas, já na parte III, houve um probleminha: eu tenho um TK83 e todos os CI's estão raspados, ficando impossível identificá-los.

Sendo assim não pude fazer a adaptação, e peço a ajuda do Bernhard para obter o número dos CI's na placa do TK83. E se for possível, publicar uma lista contendo todos os números dos CI's do TK83. Acho que estaria assim resolvendo milhares de problemas de usuários que se encontram na mesma situação em que me encontro, ou até pior. Também não entendi a expressão: Ligar ao pino 20 "levantado" da...

ERINALDO ALENCAR RODRIGUES
Fortaleza – CE

μBits: Enviamos o seu problema ao Bernhard, que prometeu uma matéria especial sobre o assunto. Aguarde...

● Solicito informações referentes à colocação dos códigos da linguagem Assembly no Jogo "Avenida II" e também explicações sobre códigos decimais e hexadecimais.

ALOISIO SÉRGIO ARAÚJO DA SILVA

Salvador – BA

μBits: Quando o assunto for Assembly, solicitamos que qualquer leitor escreva

diretamente para o endereço da Editora Campus no Rio de Janeiro, colocando no envelope:

COLUNA ALÉM DO BASIC
A/C Nelson N. S. Santos

Ele é o responsável por "tirar as dúvidas" dos nossos leitores interessados em linguagem de máquina.

● Por meio desta, estou remetendo as minhas soluções às propostas do nº 3 da MICROBITS.

Os conhecimentos de Assembly que me conduziram às soluções foram adquiridos com a leitura atenta da ótima seção "Além do BASIC" e do próprio livro, muito didático por sinal. Parabéns ao Sr. Nelson. Em breve enviarei algumas colaborações.

FRANCISCO DE ASSIS NUNES DA SILVA
São Paulo - SP

µBits: Enviamos a sua carta ao Nelson, que agradece os elogios e prometeu publicar as suas soluções no número 5 da revista - não havia tempo útil para serem colocadas já. O mesmo vale para as respostas enviadas pelo leitor Paulo Roberto de Castro, de São José dos Campos. Aguardamos as colaborações que você prometeu enviar.

● A revista MICROBITS veio preencher uma lacuna existente no mercado de revistas especializadas em microcomputadores, pois até agora não tínhamos (pelo menos aqui no Nordeste) uma publicação deste gênero, exclusiva para equipamentos de lógica SINCLAIR.

Quero parabenizá-los pelo tipo de programas predominantes no último número, e espero que continuem a publicar programas aplicativos, como o "ARQUIVO DE CUSTOS" e os dois programas para cálculo de integrais.

Sou estudante do penúltimo período de Engenharia Civil, e ando a procura de programas relativos à minha área para o meu RINGO R-470, especificamente sobre cálculo estrutural. Gostaria que se "pintasse" algum programa dos leitores sobre esse assunto vocês publicassem. Oportunamente enviarei alguns desenvolvidos por mim para vocês analisarem, e se possível publicarem.

RICARDO JOSÉ MOREIRA SOUTO
João Pessoa - PB

µBits: Agradecemos os comentários elogiosos, Ricardo. Sempre que possível publicaremos aplicativos para os nossos "pequeninos" micros. Estamos aguardando as colaborações que você vai enviar.

seção

DESAFIO

Freqüentemente é necessário obter uma tela cheia com caracteres "aleatórios". Uma maneira de obter isto em BASIC é:

```
10 RAND
20 FOR I = 1 TO 704
30 PRINT CHR$(INT(64*RND)+128*INT(2*RND));
40 NEXT I
```

Mas ... como é lento! O desafio é escrever algo em ASSEMBLY que produza um efeito inteiramente equivalente, mas rápido. A rotina que escrevemos em che a tela de maneira quase instantânea, gasta apenas 29 bytes e é chamada por RAND USR 16514.

Provavelmente existem outras soluções. Envie-nos a sua! Aqui está a resposta do DESAFIO da MICROBITS Nº 03.

TELA	16514	16	00		LD D, 0
	16516	3E	06		LD A, 6
	16518	2A	0C	40	LD HL, (D.FILE)
	16521	0E	16		LD C, 22
LINHA	16523	23			INC HL
	16524	06	20		LD B, 32
POKE	16526	77			LD (HL), A
	16527	23			INC HL
	16528	10	FC		DJNZ POKE
	16530	0D			DEC C
	16531	20	F6		JRNZ, LINHA
	16533	C6	80		ADD A, 128
	16535	15			DEC D
	16536	20	EC		JRNZ, TELA
	16538	C9			RET

No próximo número publicaremos as soluções encontradas pelos leitores FRANCISCO DE ASSIS NUNES DA SILVA e PAULO ROBERTO DE CASTRO, que são diferentes da nossa e diferentes entre si. Parabéns!

Nelson N. S. Santos

CHARADA

Não sabemos se Murphy, ao enunciar sua célebre Lei de Chateação Máxima, previu as fantásticas implicações que ela teria no campo editorial.

Oferecemos aqui a nossa modesta colaboração com os seguintes corolários da Lei de Murphy.

Corolário nº 1 - da eterna falibilidade do revisor.

O erro de impressão só pode ser visto depois de impresso.

Corolário nº 2 - da localização do erro.

Uma vez estabelecido que o erro de impressão é inevitável, ele irá localizar-se onde causar o maior dano possível.

Assim, a charada publicada na página 5 da MICROBITS ANO II - Nº 3, perguntando "O que faz este programa em linguagem de máquina?" tem uma resposta desagradável: CRASH! O programa foi publicado com um único erro, mas que o invalida completamente.

Estamos então repetindo sua listagem hexa, só que desta vez correta (esperamos...)

```
16514 2A 0C 40 44 4D 16 3B CB
16522 22 2A 10 40 0A BA 28 03
16530 C6 80 02 03 BF ED 42 20
16538 F0 C9
```

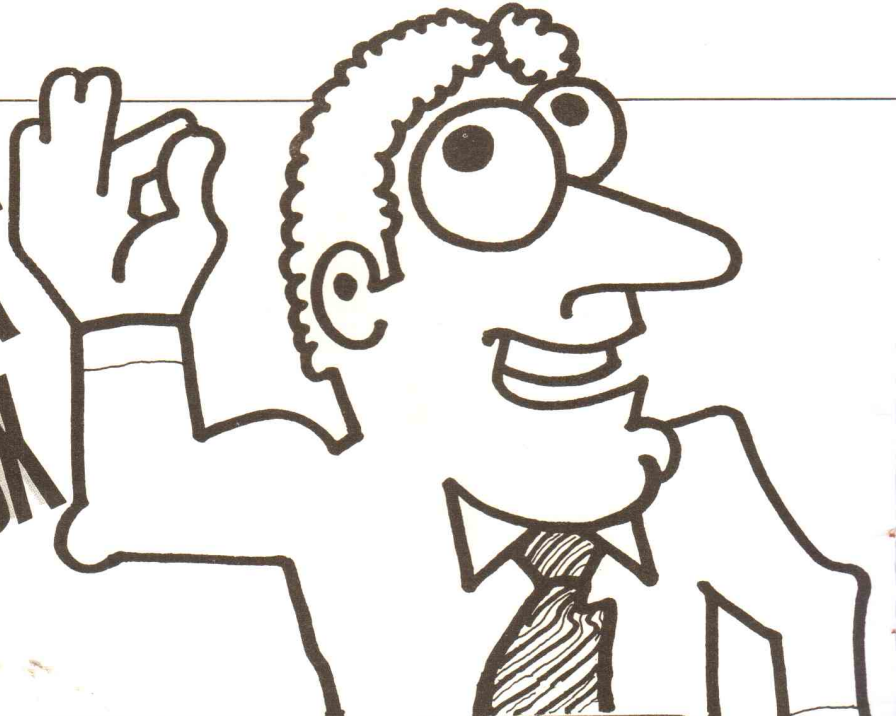
RAND USR 16514

Em tempo: meus mais sinceros parabéns ao leitor PAULO ROBERTO DE CASTRO, que "matou" a charada, apesar do erro de impressão.

ENVIEM COLABORAÇÕES!

Nelson N. S. Santos

"RAGECOLOR" PARA TK90X



O LANÇAMENTO DO TK90X É UMA GRANDE NOTÍCIA, E MICROBITS ABRE SUAS PÁGINAS PARA PROGRAMAS DESTINADOS A ESTE FANTÁSTICO MICRO.



O rigem do programa: MICROBITS Ano 1, Nº 2, página 11. A estrutura desse programa é muito conveniente para a adaptação a cores, utilizando a variável "t" (ou "T" na versão original), ligada aos números dos carros. (Obs.: poderá ser usado "T" ou "t" sem que haja alteração, pois as letras maiúsculas e minúsculas são intercambiáveis para os nomes das variáveis).

Passo a descrever as adaptações que foram feitas:

linha 30

após o ";" foi acrescentada a frase

```
FOR n = 1 TO 5: BEEP .5, n: NEXT N
```

Com isto é gerada uma escala musical que antecede a escolha do número de carros.

linha 85

a instrução PRINT obriga o próximo PRINT a ser executado na linha seguinte.

linha 90

como o "NEW LINE" no SPECTRUM é chamado "ENTER", foi feita a adaptação para: PRINT "apertar enter".

linha 94

com a instrução OVER 1, damos condição para formar o

"ç", isto é, cê-cedilha. Concordo que é um pouco complicado, mas é um recurso que computadores maiores não possuem!

linha 95

na frase — PRINT "para começ"; CHR\$ 8; " "; "ar" — a instrução CHR\$ 8 faz a imagem retornar uma casa e "imprime" a vírgula sobre o cê.

linha 96

a instrução OVER 0 "desliga" a capacidade de "overlapping".

linha 160

já nesta linha, é iniciado o processo de "pintar os carrinhos" através da instrução INK(t-1) que é repetida nas linhas seguintes. Empregamos (t-1) para evitar a cor branca. Caso contrário, o carrinho branco não iria aparecer.

linha 200

nesta linha foi feita uma modificação. Ao invés de $p(t) < 26$ escolhi $p(t) < 24$ com um resultado melhor para o SPECTRUM, pois o carrinho vencedor pára ao tocar o "█".

linha 210

nesta linha foram feitos vários acréscimos de modo a explorar a capacidade do SPECTRUM:

- Entre PRINT AT y+1,1; e ;t; foi acrescentado "→" que imprime uma flecha antes do número do carro vencedor.
- Antes de imprimir - ;"ganha";- vamos fazer esta palavra piscar em positivo e negativo na cor do carro vencedor, através das instruções - ;FLASH 1;- que liga o pisca-pisca e - ;INK(t-1); que colore a palavra com a cor do carro vencedor.
- Após a impressão da palavra *ganha*, inicia uma música que festeja o carro que venceu a corrida. Ou seja: ;:BEEP 1,-3:BEEP .5,2:BEEP .75,2:BEEP .5,6:BEEP .75,6:BEEP 1.5,9:BEEP 2,6

De resto, o programa é idêntico ao original e permanecem válidas as explicações dadas para o programa original. É interessante, para notar a diferença, digitar o programa original e depois ampliá-lo para som e cores.

Para evitar as diferenças entre letras maiúsculas e minúsculas (embora, como mencionado anteriormente, essa diferença não influa em nada), basta utilizar o comando CAPS LOCK.

Quanto aos símbolos gráficos, também existem diferenças entre o TK 82/83/85 e o ZX SPECTRUM. Para esclarecer os símbolos usados, segue uma relação dos originais e os para o Spectrum.

Programa Original	SPECTRUM
160 = "SP, SP, g8, g7, g5"	160 = "SP, SP, g5, g3, CAPS SHIFT g5"
170 = "SP, gSP, gSP, gSP, gSP, gSP, gSP" e "g4"	170 = "SP, CAPS SHIFT g8, CAPS SHIFT g8, CAPS SHIFT g8, CAPS SHIGT g8, CAPS SHIFT g8" e "CAPS SHIFT g7"
180 = "SP, O, SP, SP, SP, SP, O"	180 = "SP, CAPS SHIFT O, SP, SP, SP, SP, CAPS SHIFT O"
	g8 - significa CAPS SHIFT - tecla 9 - (cursor passa para "G") - tecla 8
	CAPS SHIFT g8 - item anterior juntamente com CAPS SHIFT
	g7 - vide g8
	SP - espaço (SPACE)

```

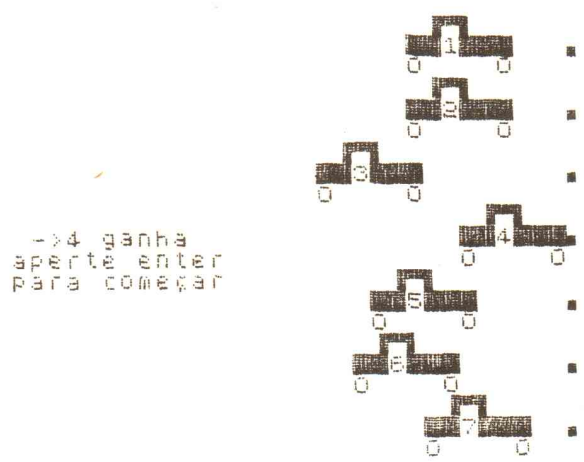
100 REM FACECOLOR para SPECTRUM
100 REM P/TK 90X digite SOUND
    SO invés de BEEP
20 REM por Alexander Gromow
30 PRINT AT 0,0;"Quantos carro
40 g 7) " : FOR n=1 TO 10: BE
P .0,n: NEXT n
40 INPUT dm
50 IF dm<2 OR dm>7 THEN GO TO
60

```

```

60 PRINT dm
70 DIM p(dm)
80 LET prim=0
85 PRINT
90 PRINT "aperte enter"
94 OVER 1
95 PRINT "para comec";CHR$ 8;"
ar"
95 OVER 0
100 INPUT i$
110 CLS
120 FOR t=1 TO dm
130 LET y=3*(t-1)
140 IF AND>.5 AND prim=1 THEN G
O TO 230
150 LET p(t)=p(t)+1
160 PRINT AT y,p(t); INK (t-1);
170 PRINT AT y+1,p(t); INK (t-1)
";" ;TAB 31);"
180 PRINT AT y+2,p(t);" 0 0"
190 PRINT AT y+1,p(t)+3;T
200 IF p(t)<24 THEN GO TO 230
210 PRINT AT y+1,1;"->";t; FLASH
H 1; INK (t-1);"ganha"; BEEP 1
.5,2; BEEP .75,2; BEEP .5,6; BEEP
.75,6; BEEP 1.5,9; BEEP 2,6
220 GO TO 70
230 NEXT t
240 LET prim=1
250 GO TO 120

```



(Nota do Editor: Este programa foi desenvolvido num ZX SPECTRUM e devem ser observadas as seguintes diferenças entre este micro e o TK 90 X:

1. O recurso utilizado para imprimir o "ç" no TK 90 X pode ser feito através de "UDG".
2. A função "BEEP" é "SOUND" no TK 90 X.)

ALEXANDER GROMOW

Alexander Gromow é engenheiro eletrotécnico e trabalha como gerente na SIEMENS S.A.

TRAÇO

Este programa traça linhas retas entre dois ou mais pontos. Você entra com uma série de coordenadas no computador que, em seguida, liga os pontos com retas.

Podem ser desenhados triângulos, quadrados etc. Caso você queira, o computador poderá criar pontos aleatórios, originando desenhos como na figura 2.

```

  T R A Ç O
  "
  
```

ESTE PROGRAMA TRACA LINHAS RETAS ENTRE DOIS OU MAIS PONTOS QUAISQUER. VOCE INFORMA AS COORDENADAS DOS PONTOS, E SERAO TRACADAS AS RETAS. SE QUISER, OS PONTOS PODEM SER ESCOLHIDOS ALEATORIAMENTE.

ESCOLHA A OPCAO VOCE INFORMA OS PONTOS OU QUER QUE SEJAM ALEATORIOS (U/A) ?

QUANTOS PONTOS ? (2-20) = 20

Figura 1 - Instruções

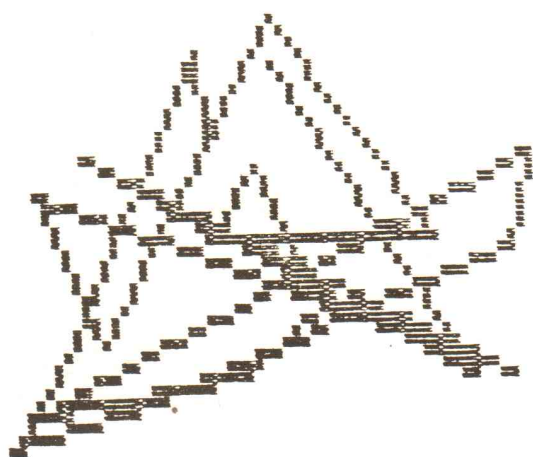


Figura 2

PROGRAMA "TRAÇO"

```

REM *****
*           T R A Ç O   1984*
*           *
*   ALEXANDRE PFEIFER   *
*   CAIXA POSTAL 242    *
*   35500 DIVINOPOLIS MG*
*   FONE: (037)221-5836 *
*           *****

1  CLS
2  SLOW
3  POKE 16416,2
5  GOTO 190
20 INPUT X$
30 IF LEN X$ < 1 OR LEN X$ > 2 THEN
  GOTO 20
40 FOR N=1 TO LEN X$
50 IF X$(N) < "0" OR X$(N) > "9" THEN
  GOTO 20
60 NEXT N
70 IF VAL X$ < 0 OR VAL X$ > LIM THEN
  GOTO 20
80 RETURN
90 PRINT AT 20,0;"ABSCISSA:??("
0-93)";AT 20,9;
95 LET LIM=60
100 GOSUB 20
110 PRINT X$;
115 IF LEN X$ < 2 THEN PRINT " "
120 LET X=VAL X$
130 PRINT AT 21,0;"ORDENADA:??("
0-43)";AT 21,9;
135 LET LIM=43
140 GOSUB 20
150 PRINT X$;
155 IF LEN X$ < 2 THEN PRINT " "
160 LET Y=VAL X$
180 RETURN
190 PRINT AT 0,9;"T R A Ç O";
AT 1,9;"T R A Ç O";AT 2,9;"T R A Ç O";
AT 3,9;"T R A Ç O";

200 PRINT
210 PRINT "   ESTE PROGRAMA TRACA LINHAS
RETAS ENTRE DOIS OU MAIS PONTOS
QUAISQUER. VOCE INFORMA AS
COORDENADAS DOS PONTOS, E SERAO
TRACADAS AS RETAS. SE QUISER, OS
PONTOS PODEM SER ESCOLHIDOS
ALEATORIAMENTE."
220 PRINT
230 PRINT "ESCOLHA A OPCAO:"
240 PRINT "VOCE INFORMA OS PONTOS
OU QUER QUE SEJAM ALEATORIOS
(U/A) ?"
250 LET U$=INKEY$
255 RAND
260 IF U$ <> "U" AND U$ <> "A" THEN
  GOTO 250
  
```



```

260 IF U$="U" THEN PRINT AT 14,
1);"032"
265 IF U$="A" THEN PRINT AT 15,
11);"LEATORES"
270 PRINT
280 PRINT "QUANTOS PONTOS ?(2-2
0) => ";
290 LET LIM=20
300 GOSUB 20
305 PRINT X$
310 LET VAL=VAL X$
320 DIM A(VAL)
330 DIM B(VAL)
340 IF U$="A" THEN GOTO 720
350 FOR K=1 TO VAL
360 PRINT AT 19,0;"PONTO N°";K
370 GOSUB 90
380 LET A(K)=X
390 LET B(K)=Y
400 IF K>1 THEN IF A(K-1)<>X OR
B(K-1)<>Y THEN GOTO 450
405 IF K<2 THEN GOTO 450
410 FOR Z=1 TO 20
420 PRINT AT 21,0;"PONTO IGUAL
AO ANTERIOR, REDIGITE"(AT 21,0);"B
ONTO IGUAL AO ANTERIOR, REDIGITE"
430 NEXT Z
440 PRINT AT 21,0;"
445 GOTO 370
450 NEXT K
460 CLS
470 FOR K=2 TO VAL
480 LET X3=A(K-1)
490 LET Y3=B(K-1)
500 LET XDIF=ABS (X3-A(K))
510 LET YDIF=ABS (Y3-B(K))
520 LET XSTEP=SGN (A(K)-X3)
530 LET YSTEP=SGN (B(K)-Y3)

```

```

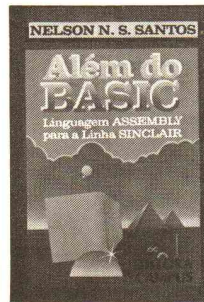
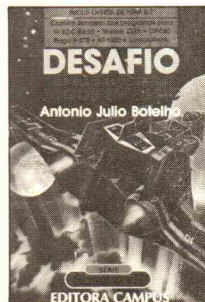
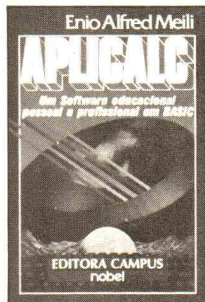
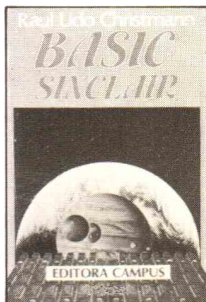
540 IF YDIF>XDIF THEN GOTO 670
550 FOR N=X3 TO A(K) STEP XSTEP
560 PLOT N,Y3
570 LET Y3=Y3+YSTEP*(YDIF/XDIF)
580 NEXT N
590 NEXT K
600 POKE 16418,0
610 PRINT AT 23,0;"COPY, OUTROS
PONTOS, REPETIR, EM ?"
615 IF INKEY$="C" THEN COPY
620 IF INKEY$="R" THEN RUN
630 IF INKEY$="F" THEN GOTO 61
5
650 CLS
660 STOP
670 FOR N=Y3 TO B(K) STEP YSTEP
680 PLOT X3,N
690 LET X3=X3+XSTEP*(XDIF/YDIF)
700 NEXT N
710 GOTO 590
720 FAST
725 FOR K=1 TO VAL
730 LET A(K)=INT (RAND*64)
740 LET B(K)=INT (RAND*44)
750 IF K>1 THEN IF A(K-1)=A(K)
AND B(K-1)=B(K) THEN GOTO 730
760 NEXT K
765 SLOW
770 GOTO 460
0096 SAVE "TRAC0"
0097 RUN
0098 RAND USR 8405
0099 RUN

```

Alexandre Pfeifer tem 15 anos e é estudante de eletrônica. Possui um TK-85 e Timex Sinclair TS 2068.

ALEXANDRE PFEIFER

OS MELHORES LIVROS NACIONAIS PARA A LINHA SINCLAIR



A EDITORA CAMPUS tem a mais completa linha de livros de computação — isto todos sabem. O que talvez você não saiba é que a CAMPUS tem a mais completa linha de livros especialmente destinados a você, usuário da linha SINCLAIR.

Se o seu desejo é:

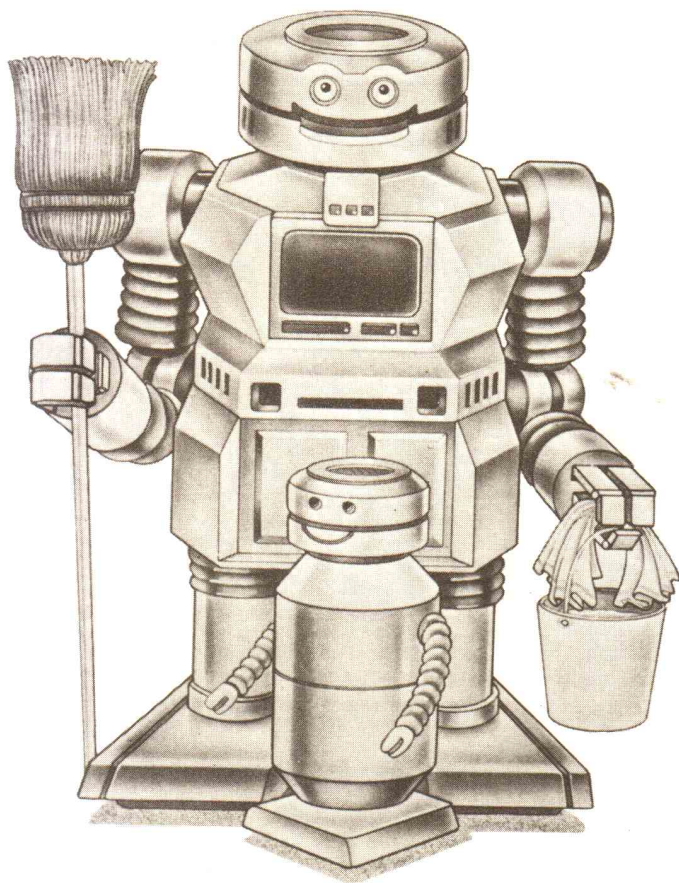
- dar os primeiros passos na aprendizagem do BASIC;
- usar profissionalmente o seu "pequeno" micro;
- divertir-se com boas horas de jogos computacionais;
- aprender linguagem de máquina;

tudo isto no seu SINCLAIR, então estes livros são indispensáveis para você. Eles podem ser encontrados nas boas Livrarias e Distribuidores. Mas, se quiser recebê-los em casa, escreva diretamente para a CAMPUS.

MAIS UM LANÇAMENTO DA

Editora Campus Ltda.

SIMULANDO



UM COMPUTADOR

O aprendizado de linguagem de máquina, diretamente em um computador, é uma tarefa ingrata, visto a dificuldade em se visualizar a execução e a possibilidade (freqüente) de um programa incorreto "travar" a máquina.

Por estes motivos, é bastante interessante o uso de um simulador, que facilite a entrada, execução e depuração de programas em linguagem de máquina.

O programa aqui descrito simula um computador hipotético (o "COMP"), permitindo a entrada de programas em "assembler" ou "linguagem de máquina", e a sua execução com a exibição na tela do conteúdo dos registradores e parte da memória.

10MICROBITS

A arquitetura do COMP

O computador COMP (ver figura 1) é composto por:

- um acumulador (AC) com capacidade de 4 dígitos decimais.
- um "indicador de sobrecarga" (C) que indica quando o resultado de uma operação não cabe no acumulador.
- um contador de programa (CP) que indica qual a posição de memória que contém a próxima instrução a ser executada.
- uma memória com 100 posições, cada uma capaz de armazenar 4 dígitos decimais. Estas posições são referenciadas através de endereços compostos de dois dígitos decimais (00 a 99).
- um dispositivo de entrada (teclado).
- um dispositivo de saída (vídeo).

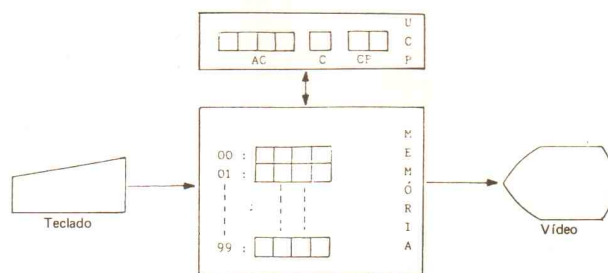


Figura 1

As instruções do COMP são codificadas em números de quatro dígitos com o seguinte formato:

IOMM

onde

O — é o código da operação

I — indica como é obtido o operando da instrução:

I = 0 — endereçamento direto, MM é o endereço do operando

I = 1 — endereçamento indireto, a posição de endereço MM contém (nos 2 dígitos menos significativos) o endereço do operando

I = 2 — imediato, MM é o próprio operando

As operações são descritas abaixo, com E indicando o endereço do operando e (E) o conteúdo de E (ou um valor imediato).

O	mnemônico	operação
0	LDA	AC ← (E)
1	STA	E ← (AC)
2	ADD	C, AC ← (AC) + (E)
3	SUB	C, AC ← (AC) - (E)
4	INP	E ← teclado
5	OUT	vídeo ← (E)
6	BRN	CP ← E
7	BZR	se (AC) = 0 CP ← E
8	BCY	se (C) = 1 CP ← E
9	STP	pára o processamento

A instrução STP não tem operando; as instruções STA, INP, BRN, BZR e BCY não permitem operando imediato.

O "Assembler" do COMP

Uma instrução do COMP é codificada em assembler da seguinte forma:

- mnemônico três letras
- separador 1 espaço
- modo de endereçamento * - indireto
(ausente se direto) \$ - imediato
- operando 1 ou 2 dígitos

Por exemplo, suponhamos que a posição de endereço 10 contenha o número 0111 e a posição de endereço 11 contenha o número 9999. Então:

- Para colocar o número 0111 no acumulador podemos usar LDA 10
- Para colocar o número 0010 no acumulador podemos usar LDA \$10
- Para colocar no acumulador o número (9999) contido na posição cujo endereço (11) está na posição de endereço 10 usamos LDA * 10

O Programa Simulador

O programa simulador é composto por duas partes:

- O "interpretador", que executa as instruções do COMP, atualizando a tela. Esta parte está programada em linguagem de máquina.
- O "monitor" que fornece os comandos para entrada, execução e monitoração de programas do COMP. Esta parte está em BASIC.

A tela está organizada em três janelas (ver figura 2):

- Uma "linha de status" que mostra os conteúdos dos registradores AC e CP, bem como o indicador C ("aceso" quando C=1) e a instrução atual IA (em código de máquina e em assembler).
- Uma imagem de 40 posições consecutivas da memória.
- Uma área onde são feitas as entradas e saídas do Monitor e do COMP

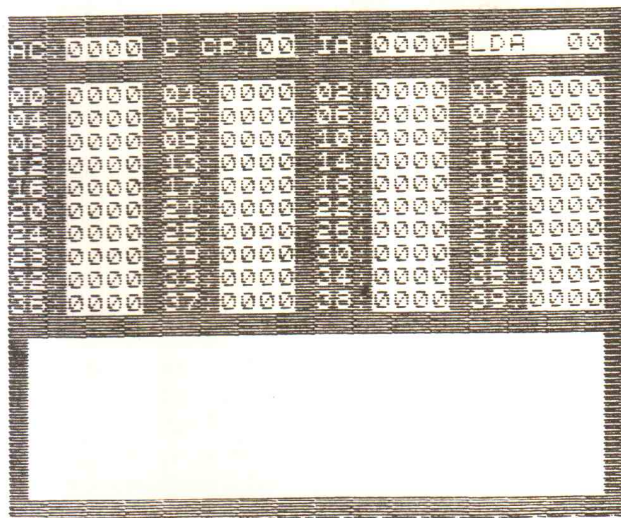


Figura 2

Os Comandos do Monitor

O monitor assinala com uma barra (/) quando está pronto para receber comandos. Estes comandos consistem de uma letra, que deve ser seguida de "Newline".

- A - Asm - Permite entrar com instruções em assembler. É pedido o endereço inicial para a colocação dos códigos (0 a 99). As instruções devem estar no formato já descrito anteriormente. A entrada de uma linha vazia encerra o comando.
- C - Clear - Zera o acumulador, o contador de programa e todas as posições da memória do COMP.
- D - Dump - Atualiza a imagem da memória na tela. É pedido o endereço da primeira posição a mostrar (0 a 60).
- E - End - Pára o programa monitor.
- L - Load - Permite entrar diretamente com números na memória. É pedido o endereço inicial (0 a 99). Os números (1 a 4 dígitos decimais) devem ser entrados um a um. A entrada de uma linha vazia encerra o comando.
- R - Run - Executa um programa. Deve-se fornecer o endereço inicial (0 a 99) e a velocidade (0 - mais rápida a 255 - mais lenta) de execução. A execução pára ao ser encontrada a instrução STP ou se pressionada a tecla SHIFT.
- S - Save - Salva em fita o programa (inclusive o estado do COMP). Convém ligar o gravador antes de dar este comando.

No caso de comando ou parâmetro inválido, o monitor coloca uma interrogação (?) e pede novo comando.

Digitação do Simulador

O primeiro passo é entrar com o interpretador (listagem 1). Para isto é necessário criar na linha 1 um REM, seguido de *exatamente* 760 caracteres. Sugere-se o uso de um monitor assembler (ver o artigo A LINHA REM, na MICROBITS Ano II, nº 3) para criar a linha REM e digitar o interpretador. Um erro nesta parte terá resultados imprevisíveis; confira-a atentamente!

Em seguida, digita-se o Monitor (listagem 2). A linha 2 é um REM com pelo menos 200 caracteres que conterá a memória do COMP.

Para salvar o programa você pode dar RUN e usar o comando S do monitor.

Dois Exemplos

O primeiro exemplo é um programa que imprime na tela os números 99, 98, 97, , 0; abaixo temos detalhadamente como entrá-lo e executá-lo:

```
D <NL> 0 <NL>
A <NL> 0 <NL>
LDA $99 <NL>
STA 10 <NL>
OUT 10 <NL>
SUB $1 <NL>
BCY 6 <NL>
BRN 1 <NL>
STP <NL>
<NL>
R <NL> 0 <NL> 20 <NL>
```



Onde <NL> significa Newline. Veja figura 3 que mostra a tela do vídeo após digitar o programa.

Experimente executar novamente à velocidade máxima: R <NL> 0 <NL> 0 <NL>

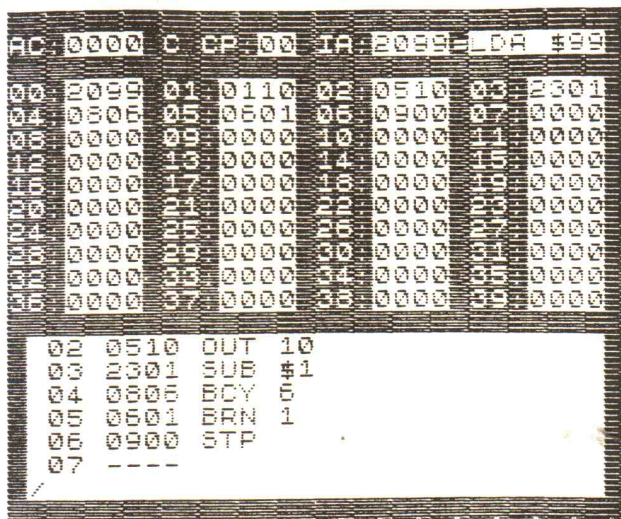


Figura 3

O segundo exemplo é dado abaixo em código de máquina. Executado por R <NL> 0 <NL> velocidade <NL> ele pedirá 6 valores e os imprimirá em ordem crescente. Observe as posições a partir do endereço 50 (D <NL> 50 <NL>) e veja porque este algoritmo é chamado "bubble sort" (ordenação por bolhas).

Listagem 3 – Exemplo 2 em linguagem de máquina.

END.	====	CONTEUDO	====
00	2050	0150 1450 0050	2201
05	0150	0355 0802 2051	0150
10	0050	0150 0050 2001	0100
15	1050	1050 0830 1050	0151
20	1050	1150 0051 1150	0050
25	2301	0150 2050 0700	0010
30	0050	2201 0150 2050	0010
35	2050	0150 1550 0050	2201
40	0150	0355 0807 0000	

Para quem prefere digitar o programa em mnemônicos, a listagem 4 mostra, em quatro colunas, o endereço, o código em linguagem de máquina e as instruções em assembler.

Listagem 4

00	2050	LDA	#50
01	0150	STA	#00
02	1450	INP	*000
03	0050	LDA	#00
04	2201	ADD	#01
05	0150	STA	#00
06	2050	SUB	#00
07	0802	BCY	#00
08	2051	LDA	#01
09	0150	STA	#00
10	0050	LDA	#00
11	0150	STA	#00

12	0050	LDA	#00
13	2301	SUB	#01
14	0100	STA	#00
15	1050	LDA	*000
16	1050	SUB	*000
17	0830	BCY	*000
18	1050	LDA	*000
19	0151	STA	*010
20	1050	LDA	*000
21	1150	STA	*000
22	0051	LDA	*000
23	1150	STA	*000
24	0050	LDA	*000
25	2051	SUB	*000
26	0100	STA	*000
27	2050	SUB	*000
28	0700	BRN	*000
29	0010	BRN	*000
30	0000	LDRN	*000
31	2201	ADD	*000
32	0100	STA	*000
33	2050	SUB	*000
34	0010	BCY	*000
35	2050	LDA	*000
36	0150	STA	*000
37	1050	OUT	*000
38	0050	LDA	*000
39	2201	ADD	*000
40	0150	STA	*000
41	2050	SUB	*000
42	0007	BCY	*000
43	0000	STP	*000

Estrutura Interna do Programa

Para quem desejar fazer alterações no programa, aqui está uma referência rápida:

- interpretador (rotinas)
 - 16516 – "scroll" da área de mensagens
 - 16519 – atualiza memória na tela
 - 16522 – atualiza "linha de status"
 - 16525 – limpa o computador
 - 16528 – inicia execução. Retorna 0 se encontrou STP, 1 se foi apertado SHIFT e 2 se encontrou INP
 - 16531 – continua execução após INP
- interpretador (variáveis)
 - 16534 – primeiro endereço de dump em binário
 - 16535 – primeiro endereço de dump em BCD
 - 16543 – velocidade de execução
 - 16544 – 2 dígitos mais significativos de valor lido para INP (em BCD)
 - 16545 – 2 dígitos menos significativos de valor lido para INP (em BCD)
 - 17280 – início da memória do COMP. Cada posição ocupa dois bytes, o primeiro com os 2 dígitos mais significativos e o segundo com os 2 menos significativos (ambos em BCD)
- monitor
 - 150 – laço principal
 - 210 – erro
 - 500 – clear
 - 1000 – load
 - 1500 – dump
 - 2000 – run
 - 2500 – asm
 - 3000 – save
 - 3500 – end
 - 4000 – ler valor
 - 4500 – inicialização


```

1110 LET ADDR=ADDR+1
1120 IF ADDR=100 THEN LET ADDR=0
1130 LET X=USR 16519
1140 LET X=USR 16522
1150 GOTO 1040
1499 REM
1500 PRINT "DUMP ";
1510 GOSUB 4000
1520 IF (N$="") OR (NBIN>60) THE
N GOTO 210
1530 POKE 16534,NBIN
1540 POKE 16535,NBCD
1550 LET X=USR 16519
1560 GOTO 150
1999 REM
2000 PRINT "RUN ";
2010 GOSUB 4000
2020 IF (N$="") OR (I<199) THE
N GOTO 210
2030 POKE 16541,NBCD
2040 LET X=USR 16522
2050 PRINT " ";
2060 GOSUB 4000
2070 IF (N$="") OR (NBIN>255) TH
EN GOTO 210
2080 POKE 16543,NBIN
2090 LET X=USR 16522
2100 IF X=0 THEN GOTO 150
2110 IF X=2 THEN GOTO 2150
2120 LET X=USR 16516
2130 PRINT AT 20,2;"** BRK **"
2140 GOTO 150
2150 LET X=USR 16516
2160 PRINT AT 20,2;"?";
2170 GOSUB 4000
2180 IF N$="" THEN GOTO 2150
2190 LET NHI=INT (NBCD/256)
2200 POKE 16544,NHI
2210 POKE 16545,NBCD-256*NHI
2220 LET X=USR 16531
2230 GOTO 2100
2499 REM
2500 PRINT "ASH ";
2510 GOSUB 4000
2520 IF (N$="") OR (NBIN>99) THE
N GOTO 210
2530 LET ADDR=NBIN
2540 LET X=USR 16516
2550 PRINT AT 20,2;CHR$(28+INT
(ADDR/10));CHR$(28+ADDR-10*INT
(ADDR/10))" ----"
2560 INPUT L$
2570 IF L$="" THEN GOTO 150
2580 PRINT AT 20,10;L$;
2590 LET L#=L$+" "
2600 LET O#=L$(1 TO 3)
2610 LET I=1
2620 IF O#=T$(I TO I+2) THEN GOT
O 2650
2630 LET I=I+3
2640 IF I>31 THEN GOTO 2620
2650 GOTO 210
2660 LET OP=(I-1)/3
2670 LET IND=0
2680 LET MEM=0
2690 IF (OP=9) AND (L$(4 TO 7)<>
") THEN GOTO 210
2700 IF OP=9 THEN GOTO 2820
2710 IF L$(4)<>" " THEN GOTO 210
2720 LET IND=1*(L$(5)="*")+2*(L$
(5)="#" )
2730 LET I=5
2740 IF IND>0 THEN LET I=6
2750 LET MEM=CODE L$(I)-28
2760 IF (MEM<0) OR (MEM>9) THEN
GOTO 210
2770 LET I=I+1
2780 IF L$(I)="" " THEN GOTO 2820

```

```

2790 LET MLO=CODE L$(I)-28
2800 IF (MLO<0) OR (MLO>9) THEN
GOTO 210
2810 LET MEM=MEM+16+MLO
2820 POKE 17280+2*ADDR,IND*16+OP
2830 POKE 17281+2*ADDR,MEM
2840 PRINT AT 20,5;CHR$(IND+28)
;CHR$(OP+28);CHR$(INT (MEM/16)
+28);CHR$(MEM-16*INT (MEM/16)+
8)
2850 LET X=USR 16522
2860 LET X=USR 16519
2870 LET ADDR=ADDR+1
2880 IF ADDR=100 THEN LET ADDR=0
2890 GOTO 2540
2999 REM
3000 PRINT "SAVE"
3010 SAVE "COM2"
3020 GOTO 150
3499 REM
3500 PRINT "END"
3510 STOP
3999 REM
4000 LET N$=""
4010 LET NBIN=0
4020 INPUT V$
4030 PRINT V$;
4040 IF V$="" THEN RETURN
4050 FOR I=1 TO LEN V$
4060 LET DIG=CODE V$(I)-28
4070 IF (DIG<0) OR (DIG>9) THEN
RETURN
4080 LET NBIN=NBIN*10+DIG
4090 NEXT I
4100 LET NBCD=NBIN+6*INT (NBIN/1
0)+96*INT (NBIN/100)+1536*INT (N
BIN/1000)
4110 LET N#=V$
4120 RETURN
4499 REM
4500 CLS
4510 FOR L=0 TO 21
4520 PRINT "
";
4530 NEXT L
4540 LET X=USR 16525
4550 PRINT AT 1,0;"[C] XXXX G EP
XX IC XXXX=XXX XXX"
4560 LET X=USR 16522
4570 POKE 16534,0
4580 POKE 16535,0
4590 LET X=USR 16519
4600 FOR L=1 TO 7
4610 LET X=USR 16516
4620 NEXT L
4630 LET T$="LDA$TARDD$SUBIN$PUTB
RNBZ$RBCY$TP"
4640 RETURN

```

Daniel Quadros é formado em Engenharia Elétrica (Eletrônica Digital) pela Escola Politécnica da USP. Trabalha na área de desenvolvimento de software da Scopus Tecnológica desde 1982. Participa, atualmente, do grupo responsável pelo sistema operacional do Nexus (SISNE). É usuário de um TK-82C, para o qual vem desenvolvendo uma série de programas.

Daniel contribui regularmente para MICROBITS desde 1983, com programas e textos sobre linguagem de máquina.

DANIEL QUADROS

ENTRADA DE LINGUAGEM DE MÁQUINA SEM O USO DO

REM

Há casos em que pretendemos colocar um determinado programa em linguagem de máquina no micro e, por horas e horas, testá-lo de várias formas com vários programas em BASIC. Porém, a cada NEW digitado, mesmo que acidentalmente, tudo vai embora e temos que entrar com todo o programa novamente.

Isto ocorre quando fazemos uso do REM, ou seja, quando "pokeamos" programas a partir do endereço 16514 e reservamos caracteres em um REM.

Neste caso devemos reservar uma área segura na memória onde o BASIC não tenha acesso. Assim, após ter sido colocado o programa em linguagem de máquina, ele só será destruído desligando-se o computador, porque um NEW não irá limpar estas posições de memória.

Para tanto, basta fazer o seguinte:

— conforme a disponibilidade de memória RAM de sua máquina, execute os seguintes comandos diretos:

1K e 2K	4K	16K
POKE 16388, 173 POKE 16389, 67 NEW	POKE 16388, 32 POKE 16389, 78 NEW	POKE 16388, 48 POKE 16389, 117 NEW

Tabela 1.

O efeito será imediato. Os endereços 16388 e 16389 pertencem à variável do sistema chamada RAMTOP. Eles fornecem o endereço do primeiro byte que o computador não pode usar, pelo menos em BASIC.

Em circunstâncias normais este endereço é aquele imediatamente após o último byte da memória, ou seja, toda a memória está disponível para o BASIC.

Com o "pokeamento" acima, alteramos este endereço para que alguma memória esteja disponível para entrada de programas em linguagem de máquina. O BASIC neste caso não irá transpor esta área, nem mesmo o comando NEW.

Os endereços que podem ser usados para programas Assembler são vários, porém devemos nos acostumar a certos valores confiáveis para fácil memorização.

Os endereços por nós adotados são aqueles da tabela 1, ou sejam:

1K/2K	17325
4K	20000
16K	30000

Veja se não é verdade:

1K/2K	$67 \times 256 + 173 = 17325$
4K	$78 \times 256 + 32 = 20000$
16K	$117 \times 256 + 48 = 30000$

Supondo-se que você já tenha um programa em linguagem de máquina em mente, digite o programa a seguir, que serve para introduzir seu programa a partir do endereço por você adotado conforme tabela 1:

```

10 LET X = (endereço conforme tabela 1)
20 LET A$ = ""
30 IF A$ = "" THEN INPUT A$
40 IF A$ = "S" THEN STOP
50 POKE X, 16*CODE A$(2) - 476
60 SCROLL
70 PRINT X, A$(TO 2)
80 LET X = X + 1
90 LET A$ = A$(3 TO)
100 GOTO 30
    
```

Agora digite RUN/NEWLINE. O computador está pronto para receber os códigos em linguagem de máquina que devem ser fornecidos em qualquer número par de caracteres de cada vez. Isto graças à linha 90, que permite a entrada de strings longos.

Na tela serão mostrados os endereços onde estarão localizados cada byte de seu programa, tornando fácil a verificação.

Ao terminar de digitar os comandos, digite "S" e NEWLINE.

Agora basta digitar um PRINT USR endereço ou RAND USR endereço, conforme tabela 1, de acordo com as características do programa.

A grande vantagem deste processo é que se agora você digita NEW/NEWLINE, o programa BASIC desaparecerá da memória, porém aquele em linguagem de máquina permanecerá.

Digite novamente PRINT ou RAND USR endereço e você verá executada mais uma vez a rotina em linguagem de máquina.

JOGO DOS BURACOS

A finalidade deste jogo é que você e o micro apaguem, sucessivamente, os buracos mostrados na Figura 1. Pode-se apagar quantos desejar na mesma linha, e quem apagar o último buraco será o perdedor.

Quando iniciado o jogo, você decide quem começará — você ou o micro. Quando for a sua vez você deverá escolher qual a linha que deseja, e quantos buracos quer deixar remanescendo na linha (veja Figura 2 — um instante no meio do jogo). Qualquer jogada inválida será informada pelo micro.

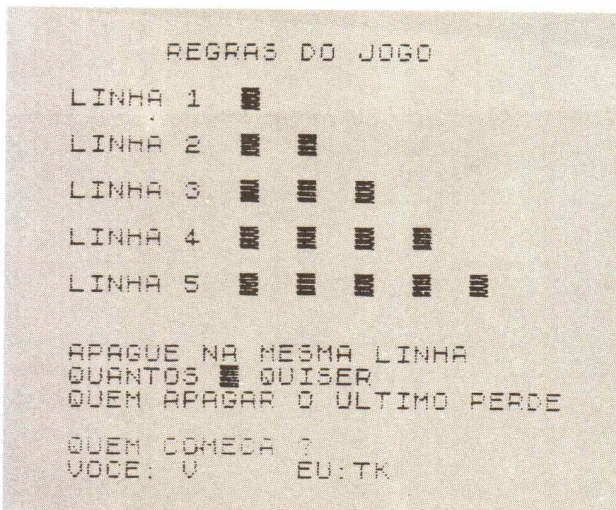


Figura 1: Início do jogo

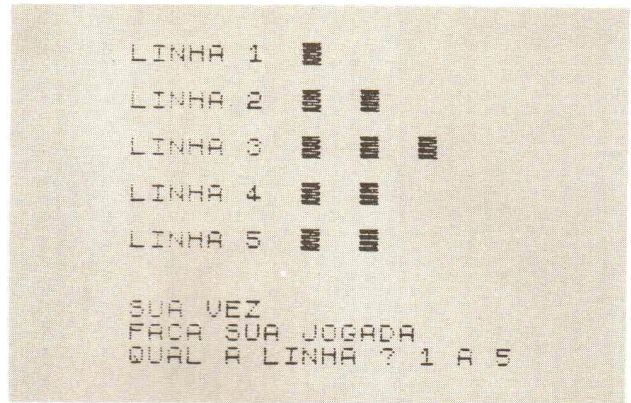


Figura 2: Um instante no meio do jogo

```

1 REM R. COLONNA
2 REM JOGO DOS ■
5 LET I=0
10 DIM J(5)
15 FAST
20 FOR F=1 TO 5
25 LET J(F)=F
30 NEXT F
35 PRINT AT 2,5;"REGRAS DO JOG
0"
40 PRINT
45 GOSUB 300
75 PRINT
80 PRINT "APAGUE NA MESMA LINH
A"
85 PRINT "QUANTOS ■ QUISER"
90 PRINT "QUEM APAGAR O ULTIMO
PERDE"
95 PRINT
100 PRINT "QUEM COMECA ?"
105 PRINT "VOCE: V      EU:TK"
106 SLOW
110 INPUT Q$
115 IF Q$(1)<>"V" AND Q$(1)<>"TK" THE
N GOTO 900
120 IF Q$="TK" THEN GOTO 400
125 CLS
130 GOSUB 300
134 PRINT
135 PRINT "FAÇA SUA JOGADA"
140 PRINT "QUAL A LINHA ? 1 A 5
"
145 INPUT L
150 IF L<1 OR L>5 THEN GOTO 140
155 LET L=INT L
165 PRINT "QUANTOS ■ V. DEIXA N
A LINHA "L";" ?"
170 INPUT M
175 LET M=INT M
180 IF M<0 OR M>=J(L) THEN GOTO
900
185 LET J(L)=M
190 CLS
195 GOSUB 300
200 PRINT
205 PRINT "AGORA SOU EU"
210 FOR F=1 TO 40
215 NEXT F
220 LET I=1
225 GOTO 400
300 FOR F=1 TO 5
305 PRINT "LINHA ";F;"  ";
310 FOR G=1 TO J(F)
315 PRINT "■ ";
320 NEXT G

```



```

325 PRINT
326 PRINT
330 NEXT F
335 RETURN
400 IF I=0 THEN GOTO 300
405 FAST
410 CLS
415 GOSUB 700
420 IF S=0 THEN GOTO 950
422 IF S<=5 THEN GOTO 1100
425 GOSUB 500
430 IF A/2=INT (A/2) AND B/2=IN
T (B/2) AND C/2=INT (C/2) THEN G
OTO 730
435 FOR H=1 TO 5
440 LET Y=J(H)
442 FOR P=1 TO 5
445 LET J(H)=J(H)-P
450 IF J(H)<0 THEN GOTO 470
455 GOSUB 500
460 IF A/2=INT (A/2) AND B/2=IN
T (B/2) AND C/2=INT (C/2) THEN G
OTO 650
470 LET J(H)=Y
472 NEXT P
475 NEXT H
500 LET A=0
505 LET B=0
510 LET C=0
515 FOR L=1 TO 5
520 GOSUB (500+10*J(L))
525 NEXT L
530 RETURN
600 REM
605 RETURN
610 LET C=C+1
615 RETURN
620 LET B=B+1
625 RETURN
630 LET C=C+1
631 LET B=B+1
635 RETURN
640 LET A=A+1
645 RETURN
650 LET A=A+1
651 LET C=C+1
655 RETURN
700 LET S=0
705 FOR F=1 TO 5
710 LET S=S+J(F)
715 NEXT F
720 RETURN
730 LET X=INT (RND*5)+1
735 IF J(X)=0 THEN GOTO 730
740 LET J(X)=J(X)-1

```

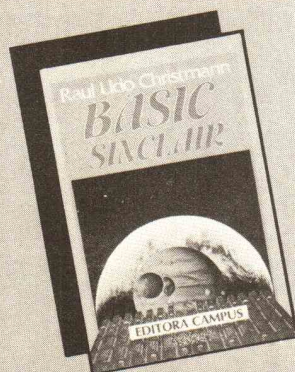
```

745 GOSUB 700
750 IF S=0 THEN GOTO 375
755 GOSUB 300
760 SLOW
765 GOTO 135
800 LET I=INT (RND*3)+1
805 GOTO (800+10*I)
810 LET J(1)=0
815 GOTO 850
820 LET J(3)=2
825 GOTO 850
830 LET J(5)=4
850 CLS
855 GOSUB 300
860 PRINT
865 PRINT "SUA VEZ"
870 SLOW
875 GOTO 135
900 CLS
905 PRINT "ATENCAO***"
910 GOTO 95
930 CLS
935 PRINT "JOGADA INVALIDA***"
940 PRINT
945 GOTO 130
950 PRINT AT 2,5;"VOCE PERDEU"
955 PRINT
960 PRINT "VAMOS JOGAR OUTRA ?"
961 PRINT "DIGITE QUALQUER LETR
A:"
965 INPUT Z#
968 CLS
970 RUN
975 PRINT AT 2,5;"VOCE GANHOU.
PARABENS"
980 SLOW
985 GOTO 955
1070 LET J(T)=1
1075 GOTO 850
1100 LET U=0
1105 FOR W=1 TO 5
1110 IF J(W)>1 THEN LET U=U+1
1115 NEXT W
1120 IF U>=2 THEN GOTO 425
1125 IF U=0 THEN GOTO 730
1130 FOR W=1 TO 5
1135 IF J(W)>1 THEN LET Y=J(W)
1140 IF J(W)>1 THEN LET T=W
1145 NEXT W
1150 IF (S-Y)/2=INT ((S-Y)/2) TH
EN GOTO 1070
1155 LET J(T)=0
1160 GOTO 850
1190 SAVE "ROG 8"
1195 RUN

```

ROGÉRIO COLONNA DOS SANTOS

PRIMEIRO FOI O
BASIC BÁSICO
 DEPOIS, O
BASIC PARA MICROS
 AGORA,
 especialmente para os
 leitores de MICROBITS:
BASIC SINCLAIR
 Christmann



- Primeira obra especificamente destinada aos possuidores e usuários de micros da linha Sinclair.
- Assuntos abordados em ordem crescente de dificuldade.
- Numerosos exemplos, dicas e problemas.
- Seção "Desafio" ao final dos capítulos.
- Apêndices úteis, complementando o texto com um resumo de todas as funções e comandos, roteiro para entender e contornar erros de sintaxe, glossário dos termos técnicos de uso corrente, etc.

MAIS UM LANÇAMENTO DA EDITORA CAMPUS LTDA.

ALÉM DO BASIC

Já temos o direito de dizer:
linguagem de máquina tratada de maneira simples e
compreensível é com a MICROBITS.



Neste nosso terceiro encontro, antes da conversa habitual sobre linguagem de máquina, gostaria de tornar maior nosso contato e ampliar os limites desta coluna, através da criação do SERVIÇO DE CONSULTA MICROBITS.

Recorte, tire uma xerox ou simplesmente copie a FICHA DE CONSULTA publicada no pé desta página, preencha seus dados e envie para nós a sua dúvida sobre programação Sinclair. Faremos o maior esforço possível para esclarecer você através da revista. Só há uma forma da MICROBITS ser melhor do que é: *voce participar dela*.

Importante: enderece-a para:

EDITORA CAMPUS LTDA.
Serviço de Consulta MICROBITS
Rua Barão de Itapagipe, 55
20261 Rio de Janeiro - RJ

Agora, direto ao assunto... Na MICROBITS Ano II - Número 2 foi publicado, na Seção "DICAS", um programa de Manoel Silva Rodrigues: "Usando o TK como máquina de escrever". Vamos aproveitar este tema e apresentar - explicando tudo detalhadamente - um programa "máquina de escrever".

O mais simples dos programas BASIC que faz isto é:

```
10 IF INKEY$ <> "" THEN GOTO 10
20 IF INKEY$ = "" THEN GOTO 20
30 PRINT INKEY$;
40 GOTO 10
```

A linha 10 espera você retirar o dedo do teclado, e a linha 20 espera você pressionar uma tecla.

Como fazer isto em linguagem de máquina? Há duas sub-rotinas da ROM que precisamos conhecer. A primeira delas é KEYBOARD SCAN que, como seu nome indica, varre o teclado verificando se houve alguma tecla pressionada. Seu endereço é 699 (02BB).

Após a chamada desta sub-rotina (CALL KEYBOARD SCAN = CD BB 02), que destrói - como quase todas as sub-rotinas da ROM - o conteúdo prévio de todos os

registradores, a informação retorna no par de registradores HL.

Se não houver tecla pressionada, HL retorna com FFFF. Havendo alguma tecla pressionada, o valor de HL é diferente. Por exemplo, o acionamento da tecla A sem shift produz em HL o valor FDFD, e o acionamento da tecla B com shift (correspondente a *) produz em HL o valor DE7F.

Precisamos primeiramente implementar a instrução:

```
10 1F INKEY$ <> "" THEN GOTO 10
```

que aguarda você retirar o dedo da tecla.

Se existir uma tecla pressionada, L não conterá FF, logo INC L não produzirá 00 e devemos retornar ao começo. Entendido?

INÍCIO	CALL KEYBOARD SCAN	CD BB 02
	INC L	2C
	JR NZ, INÍCIO	20 FA

Agora vamos ver como implementar a instrução:

```
20 IF INKEY$ = "" THEN GOTO 20
```

que aguarda você pressionar uma tecla.

Não existindo tecla pressionada, L conterá FF, e assim INC L produzirá 00, e devemos continuar esperando. Este é o esquema:

ESPERA	CALL KEYBOARD SCAN	CD BB 02
	INC L	2C
	JR Z, ESPERA	28 FA

Desta maneira conseguimos o efeito de espera! Mas, para poder imprimir na tela o caractere correspondente à tecla pressionada, necessitamos outra sub-rotina da ROM, a FIND CHARACTER, de endereço 1981 (07BD). Esta sub-rotina associa o valor produzido por KEYBOARD SCAN ao endereço do caractere - com ou sem shift - correspondente à tecla pressionada. Verifique se você entendeu.

SERVIÇO DE CONSULTA MICROBITS

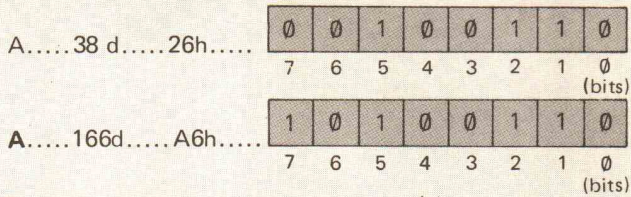
Nome.
Endereço. Nº
CEP. Cidade
Estado UF

INFORMAÇÕES SOLICITADAS

Anote no espaço acima o nome das matérias, artigos, dicas e programas sobre os quais você desejaria receber informações, seguido da especificação de sua consulta.

Tecla Pressionada	HL (por KEYBOARD SCAN)	HL (por FIND CHARACTER)
A sem shift B com shift	FDFD DE7F	0082 (130d) 00CB (203d)

Veja por exemplo os caracteres A e A :



Para testar se o bit 6 do acumulador é 0 (caractere imprimível), usaremos:

BIT 6, A CB 77

que afeta a ZERO FLAG.

Voltaremos ao BASIC caso o caractere não seja imprimível.

RET NZ C0

Agora vamos imprimir o caractere

RST 10h D7

E começar tudo de novo!

JR INICIO 18 E8

Aqui está a listagem hexadecimal completa do nosso curtíssimo programa: apenas 24 bytes.

16514	CD	BB	02	2C	20	FA	CD	BB
16522	02	E5	C1	2C	28	F8	CD	BD
16530	07	7E	CB	77	C0	D7	18	E8

O entendimento deste programa é essencial à programação de jogos em linguagem de máquina, pois permite INKEY\$. Até a próxima!

RESPOSTAS DOS EXERCÍCIOS DA COLUNA ANTERIOR

EX. 01 POKE 16519,151

EX. 02 POKE 16518,23
POKE 16519,3
POKE 16522,151

Não, isto não poderia ser feito diretamente em BASIC – necessitaria ser precedido de POKE 16418,0.

EX. 03

a) O caractere de construção da diagonal passa a ser *, pois a instrução rotulada "POKE" passa a ser LD(HL),23.

b) Em vez de uma diagonal, obtém-se uma coluna com 22*, pois a instrução LD DE,34 tornou-se LD DE,33. Somar 33 equivale a passar para a posição imediatamente abaixo.

Algumas observações importantes:

a. A tabela de caracteres "normais" (sem shift) vai do endereço 126 (007E) até 164 (00A4) e a tabela dos caracteres com shift vai de 165 (00A5) até 203 (00CB).

b. Se desejar apenas comprovar os exemplos acima, digite (modo direto)

PRINT PEEK 130, CHR\$ PEEK 130 para obter 38 e A
PRINT PEEK 203, CHR\$ PEEK 203 para obter 23 e *

c. Para comprovar a tabela toda, rode o seguinte programa:

```
10 FOR E = 126 TO 203
20 SCROLL
30 PRINT E; "-----"; PEEK E, CHR$ PEEK E
40 NEXT E
```

Para chamar FIND CHARACTER, é necessário cumprir o seguinte ritual:

- Chamar KEYBOARD SCAN;
- Transferir o valor de HL para BC;
- Chamar FIND CHARACTER, que fornecerá o endereço do caractere correspondente à tecla pressionada em HL.

Assim, vamos alterar a parte "ESPERA" do nosso programa.

```
ESPERA CALL KEYBOARD SCAN    CD BB 02
        PUSH HL                E5
        POP BC                 C1
        INC L                   2C
        JR Z, ESPERA           28 F8
        CALL FIND CHARACTER    CD BD 07
```

PUSH HL seguido de POP BC transfere o valor obtido através de KEYBOARD SCAN de HL para BC, para permitir a chamada de FIND CHARACTER. Se não houve tecla pressionada, HL contém FFFF. Assim INC L produz 00 e voltamos ao rótulo "ESPERA".

Ao ser pressionada uma tecla, obtemos em HL o endereço do caractere correspondente. Vamos obter o conteúdo deste endereço no acumulador.

```
LD A,(HL)        7E
```

Precisamos agora saber se este caractere é imprimível, para que possamos usar RST 10h para colocá-lo na tela. Como fazer isto?

Os caracteres imprimíveis dos Sinclair vão de 0 a 63 (00 até 3F) e de 128 a 191 (80 até BF). Em binário, tais números têm a seguinte particularidade: o bit 6 é sempre 0. Usando o jargão: o bit 6 é ressetado.

- c) Obtém-se a primeira linha da tela parcialmente preenchida com 22*, pois a instrução LD DE,33 tornou-se LD DE,1. Somar 1 equivale a passar para a posição imediatamente ao lado.
- d) Obtém-se uma diagonal com 24*, pois a instrução LD DE,34 foi recuperada, mas o contador B foi carregado com 24.
- e) um lindo CRASH, pois o NEW LINE de fim da primeira linha seria substituído por um*.

EX. 04

	Ø REM 17 caracteres quaisquer	
	LD HL,(D.FILE)	2A 0C 40
	LD C,22	0E 16
LINHA	INC HL	23
	LD B,32	06 20
POKE	LD (HL),136	36 88
	INC HL	23
	DJNZ POKE	10 FB
	DEC C	0D
	JR NZ,LINHA	20 F5
	RET	C9

EX. 05

	Ø REM 20 caracteres quaisquer	
	LD HL,(D.FILE)	2A 0C 40
	LD C,22	0E 16
LINHA	INC HL	23
	LD B,32	06 20
POKE	LD (HL),Ø	36 00
	INC HL	23
	DJNZ POKE	10 FB
	DEC C	0D
	JR NZ,LINHA	20 F5
	CALL PRINT AT	CD F5 08
	RET	C9

EX. EXTRA

	Ø REM 20 caracteres quaisquer	
	LD BC, ... (B=9 e C=Ø)	01 00 09
	CALL PRINT AT	CD F5 08
	LD A,Ø	3E 00
PRINT	RST 10h	D7
	INC A	3C
	CP 64	FE 40
	JR NZ,TESTE	20 01
	ADD A,A	87
TESTE	CP 192	FE C0
	JR NZ,PRINT	20 F5
	RET	C9

E vamos nós outra vez! Ainda é possível reduzir um byte, passando para 19 bytes! Sugestão: troque LD A,Ø por uma instrução de apenas um byte que faz o mesmo efeito. Quem se habilita?

NELSON N. S. SANTOS

ESTE LIVRO MARCA UMA NOVA ETAPA NO CARTUNISMO BRASILEIRO: O HUMOR CIBERNÉTICO



Este não é um manual para conserto de computadores.
É uma coleção de charges (123 páginas) que primeiro farão você RIR... e depois PENSAR.
Se desejar recebê-lo autografado pelo autor, recorte ou tire xerox do cupom abaixo.

SIM, desejo receber o livro **O COMPUTADOR ENGUIÇOU** autografado por seu autor, **GABOR GESZTI**.

- Mediante envio de cheque no valor de Cr\$ 22.000, nominal à **Editora CAMPUS** (o porte é por nossa conta e seu livro chega mais rápido).
- Mediante pagamento contra recebimento do livro pelo reembolso postal, acrescido das despesas postais.

MAIS UM LANÇAMENTO

Editora Campus

MICROBITS 21

COMO AUMENTAR O VOGABULÁRIO

Nos micros com a lógica Sinclair, como é o caso dos TK82, TK83, TK85, NE-Z 8000 e CP-200, o BASIC oferece poderosos comandos para o usuário, dando todas as possibilidades para escrever qualquer tipo de programa.

Confrontando este BASIC com o BASIC de outros computadores, sente-se a falta de certos comandos úteis, como o READ, DATA, RESTORE, AUTO, LEFT\$ etc., para citar apenas alguns. Todavia, a maioria destes comandos pode ser "traduzida" para o BASIC do nosso micro, usando comandos auxiliares ou — em último caso — rotinas escritas em linguagem de máquina.

Este último recurso permite ao usuário criar funções e comandos, ora inexistentes em microcomputadores, por se tratar, na maioria dos casos, de rotinas específicas e não de uso geral. Como exemplo podemos citar as rotinas do High-Speed, que dificilmente são encontradas desta maneira nos demais computadores.

A desvantagem principal das rotinas adicionais mencionadas é o fato de que a única maneira de usá-las é com auxílio doUSR, seguido de um número correspondendo ao endereço inicial desta rotina na memória. Além do mais ainda existe o inconveniente de precisar colocar estas rotinas numa área protegida da RAM, seja numa linha REM ou acima do RAMTOP. Ultimamente, muitos usuários já estão colocando estas rotinas adicionais em memórias fixas (EPROM), deixando-as à disposição do usuário a qualquer momento.

De qualquer maneira, é preciso decorar (ou manter uma relação à mão) os endereços individuais para cada rotina.

Baseando-me nas dificuldades acima descritas, procurei uma maneira de converter estes endereços em palavras BASIC, facilmente lembradas. Imagine, por exemplo, usar ao invés do comando RAND USR 8405 (referente ao SAVE em High-Speed), simplesmente a palavra HSAVE, e o seu micro executa esta rotina normalmente ...

Há duas alternativas para conseguir este resultado. A primeira — num estilo profissional — montei há muitos meses atrás, com sucesso. Para isto porém é necessário alterar o monitor do seu micro. Hoje, após análise dos resultados, comparando-os com a mão-de-obra empregada, cheguei à conclusão de que não vale a pena. De qualquer modo, para quem gosta de pesquisar nesta área relato, a seguir, os principais passos a serem observados para se obter o resultado desejado.

1. Em primeiro lugar, escolhi um comando "sem função", o REM, e alterei esta palavra para "DEF" (define...). Esta alteração pode ser feita no monitor, nos endereços a seguir:

<i>endereço decimal</i>	<i>valor original</i>	<i>valor novo</i>
407	55 (37h) = R	41 (29h) = D
408	42 (2Ah) = E	não altera = E
409	178 (B2h) = M	171 (ABh) = F

Pronto. A esta altura, cada vez que se digita a palavra REM, aparecerá na tela a palavra DEF. Todavia, o DEF continua a ser interpretado como um REM normal, sem qualquer alteração do funcionamento normal do micro.

A idéia é escrever após a palavra REM (agora aparecendo como DEF) simplesmente o comando BASIC criado por nós, por exemplo, o HSAVE. Assim sendo, ao invés de escrever RAND USR 8405, poderemos simplesmente digitar DEF HSAVE, e o micro já executa o SAVE em High-Speed.

Até agora não fizemos nada mais do que alterar a palavra REM. Resta portanto ainda alterar a rotina de sintaxe do monitor. Se não fizermos isso, o micro nunca irá analisar o que venha a ser escrito após o DEF, pois para ele ainda não passa de um simples comentário dentro do programa.

DO BASIC NO SEU MICRO

BERNHARD WOLFGANG SCHÖN

Analisando o monitor, podemos encontrar na rotina de decodificação dos comandos BASIC também a parte que trata do REM em si. Neste local há, falando globalmente, um simples RETURN, sem nenhuma análise adicional. Como não temos mais espaço no monitor, teremos que instalar um "desvio" para fora do monitor, por exemplo, para o endereço 10240 (2800h), que corresponde ao primeiro endereço após a EPROM do High-Speed no seu micro (não importa se o seu micro tem o High-Speed ou não).

Alteramos portanto o endereço de desvio da seguinte forma no monitor:

endereço decimal	valor original	valor novo
3189	106 (6Ah)	0 (00h)
3190	13 (0Dh)	40 (28h)

Agora, ao digitar o DEF, o seu micro entrará em crash, pois ainda não instalamos uma memória ROM ou EPROM no endereço 10240.

No meu projeto, instalei uma verificação de sintaxe, semelhante a já existente para o BASIC normal do micro. Também darei as dicas para esta função mais adiante. Em primeiro lugar, há necessidade de "ler" o que foi escrito após a palavra REM, já que o monitor foi desviado para o novo endereço 10240. Esta parte é mais simples do que parece.

Após o desvio, o par de registradores HL contém o endereço do primeiro caractere após o DEF, e basta fazer uma leitura para comparar este caractere com o primeiro caractere do nosso novo comando.

Para ilustrar melhor, digamos que queremos ter os comandos

HLOAD (para ler em High-Speed)
 HSAVE (para gravar em High-Speed)
 HVERIFY (para verificar)

pré-programados. Neste caso é preciso testar apenas o primeiro e o segundo caracteres para distinguir estes três comandos. Qualquer outro comando, iniciando com a letra "H", deverá resultar num erro de sintaxe.

Antes da rotina em si, para ler o próximo caractere, deve-se usar os mnemônicos RST 32, seguido de RST 24 (E7h e DFh respectivamente), que são sub-rotinas prontas no monitor usadas para interpretar a listagem do seu programa.

Assim sendo, a partir do endereço 10240, teremos a seguinte rotina decodificadora:

```
10240 LD A, (HL) 7E      carrega o valor do primeiro
                        caractere no acumulador A
                        verifica se é um "H"
CP "H"  FE 2D          se não for, retorne ao monitor
RET NZ  C0             para continuar o programa
```

```
RST 32  E7           lê o próximo caractere
RST 24  DF
CP "L"  FE 31       o próximo caractere é
                        "L"?
JP Z, xxxx CA xx xx se for, vai para o endereço
                        xxxx
CP "S"  FE 38       o próximo caractere é
                        "S"?
JP Z yyyy CA yy yy se for, vai para o endereço
                        yyyy
CP "V"  FE 3B       o próximo caractere é
                        "V"?
JP Z zzzz CA zz zz se for, vai para o endereço
                        zzzz
RST 8   CF          erro de sintaxe
código  10
```

Analisando esta rotina, podemos verificar que a rotina de erro somente será "acessada", quando a palavra após o DEF começa com um "H", mas não é seguida por um L, S ou V. Logo no início da rotina, quando a primeira letra não for um "H", haverá, então, um simples "return" ao monitor, sem qualquer alteração do funcionamento normal do micro.

Restam, portanto, ainda os endereços xxxx, yyyy e zzzz, que são os endereços iniciais das funções do HLOAD, HSAVE e HVERIFY, respectivamente. Dentro deste critério é possível criar mais comandos de BASIC para seu micro. Ainda analisando a rotina acima, pode-se observar que não é preciso escrever o novo comando BASIC na sua extensão. Basta escrever HL ao invés de HLOAD, HS ao invés de HSAVE e HV ao invés de HVERIFY, já que estamos analisando apenas os primeiros dois caracteres no exemplo acima.

De qualquer modo, a alternativa acima descrita pode ser melhorada, ampliada à vontade (no meu projeto usei 15 novos comandos), mas creio que diante da segunda alternativa, o acima descrito não passa de uma mera informação e curiosidade.

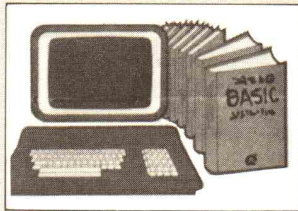
Alternativa 2:

Substituindo os endereços iniciais das rotinas escritas em linguagem de máquina por palavras que, em princípio, poderá ser feito em BASIC, iniciando todo o programas com linhas como:

```
10 LET HSAVE = 8406
20 LET HLOAD = 8630
   etc.
```

usando depois

```
RAND USR HSAVE
RAND USR HLOAD
   etc.
```

Esta solução, porém, tem várias desvantagens ponderantes:

- aumenta consideravelmente o tempo para digitar um programa, além de ocupar muita memória.
- após um NEW, RESET ou CLEAR perdem-se todas as variáveis e é preciso definir tudo novamente.

Para ganhar tempo, podemos escrever uma rotina em linguagem de máquina que define todas as variáveis desejadas numa fração de segundos. Esta rotina deve ser fixa numa EPROM, localizada, por exemplo, no endereço 10240 do seu micro. Assim, basta dar um RAND USR 10240 para fixar todas as variáveis, e repetir este comando após cada NEW, RESET ou CLEAR.

Para o programador mais ousado, vai uma dica: a rotina do CLEAR localizada no monitor a partir do endereço 140Ah, e sempre usada após o CLEAR em si e após o NEW (e RESET), podendo ser alterada para chamar por sua vez - após o CLEAR em si - a sub-rotina em 10240. Desta forma, o seu micro fixará automaticamente todos os novos comandos, atribuindo a eles todos os endereços usados.

Rotina CLEAR original no monitor:

```
149A LD HL, (4010)
      LD (HL), 80
      INC HL
      LD (4014), HL
      LD HL, (4014)
      LD (401A), HL
      LD (401C), HL
      RET
```

Rotina CLEAR alterada no monitor:

```
149A LD HL, (4010)
      LD (HL), 80
      INC HL
      LD (4014), HL
      LD HL, (4014)
      CALL 10240 - desvio para o endereço
                  10240
      LD (401C), HL
      RET
```

É importante observar que - empregando esta solução - deve-se ter em mente que os valores a serem colocados em 401A, 401C e 4014 podem mudar de acordo com a sua rotina em 10240. O que vale é a idéia e, tenho certeza, programando com cuidado obteremos os resultados desejados e um micro muito mais poderoso.

ERRATA

- No programa "TRON" publicado na edição número 3, houve um erro na linha 340 que deveria ser:

```
340 IF PEEK A=X THEN GOTO 630
```

- Em "TORRES DE HANÓI" (MICROBITS Nº 3) a linha 5550 deveria ser:

```
5550 LET C$=" " (6 espaços)
```

MINI-MÁQUINA DE ESCREVER

• Ela cabe sem problemas em qualquer pasta de executivo: a nova máquina de escrever da Olympia tem o tamanho de uma caixa de compassos e é dobrável. Tem um pequeno teclado e um monitor retangular, no qual aparece o que está sendo escrito. A pequena máquina de escrever é, ao mesmo tempo, um pequeno computador, que pode memorizar mais de 7.000 caracteres, ou seja, cerca de cinco a sete páginas de papel-ofício. Através de teclas especiais podem-se "chamar" expressões que são usadas com frequência, cortar ou acrescentar palavras ou frases inteiras. E o texto pode ser transmitido por telefone a outro aparelho do mesmo tipo. Finalmente, a máquina de escrever pode ser acoplada a uma impressora que imprime vinte linhas por minuto num rolo estreito de papel.

CONTROLE REMOTO

• Em Hamburgo já começou o futuro para o metrô; os primeiros trens experimentais rodaram ali um total de um milhão de quilômetros controlados por um computador, sem maquinista para dirigi-los. No chamado "Push-System", cada composição do metrô está registrada no computador da central através de sua designação numérica. O computador dirige o metrô no seu roteiro pré-determinado, mas analisa também a plataforma mais favorável para a parada do trem e como ele deve ser conduzido no futuro. Mas tampouco é inteiramente dispensável a maquinista do metrô, mas pode intervir a qualquer momento, caso aconteça algum imprevisto no percurso. O metrô inteiramente automático objetiva uma redução de custos dos transportes urbanos, atualmente deficitários.

LANÇAMENTO DO TK90X

• Após dois anos de desenvolvimento, a um custo estimado em US\$5.000.000 (segundo o fabricante), a Microdigital lançou seu mais recente microcomputador pessoal, o TK90X.

O TK90X, que tem aspecto similar ao TK85, embora possua um número maior de recursos, ainda oferece a vantagem de ser compatível com o microcomputador de maior sucesso no mundo, o Sinclair ZX Spectrum, que alcançou a cifra de 5 milhões de vendas.

Seguindo esta mesma linha, a Microdigital desenvolveu o TK90X com características que permitem a utilização das centenas de programas criados para o ZX Spectrum, entre jogos, utilitários e aplicativos profissionais, comerciais e educacionais.

A programação do TK90X pode ser feita utilizando o interpretador BASIC residente na ROM de 16K, em linguagem de máquina, em LOGO e em FORTH.

Com 90 comandos e instruções, controles de monitor, gravador e periféricos, o TK90X possui caracteres maiúsculos e minúsculos, além de acentuação em português e espanhol, utilizando o UDG ("User Defined Graphics").

Com seu novo micro, a Microdigital pretende, até o final de 1986, conquistar 200 mil novos usuários, além de escolas públicas e privadas, que serão grandes consumidores deste equipamento no Brasil, segundo o Diretor de Marketing, Jorge Luís dos Santos.

Características Técnicas

- Microprocessador Z80Z - 3,58 MHz
- 16K de ROM, contendo o interpretador BASIC e controles do monitor, do gravador e periféricos
- 16K ou 48K de RAM
- Cursor de reportagem: indicação de erro em português
- Cursor de programa: >
- Cursor de edição: K, L, C, E, G
- Programável em linguagem de má-

quina usando as funções: PEEK, POKE, USR, SAVE CODE, LOAD CODE, VERIFY CODE, IN, OUT, BIN

- Teclado tipo QWERTY com auto repetição e identificação sonora de aceitação de tecla. Possui 40 teclas equivalentes a funções e comandos em BASIC; 37 funções matemáticas, lógicas e científicas; 36 caracteres alfanuméricos (26 caracteres alfabéticos minúsculos e maiúsculos); 8 caracteres gráficos e 8 caracteres gráficos invertidos.
- UDG 0/1 ("User Defined Graphics") com 38 caracteres acentuados para os idiomas português e espanhol.
- UGD 2 - 21 caracteres gráficos criados pelo usuário através de Editor de UDG (matriz 8 x 8)
- Saída para vídeo; aceita conexão com TV colorida ou P/B, sistema PAL-M no canal 3; trabalha em MODO INVERTIDO controlado por software; som modulado pela TV; display de texto de 24 linhas por 32 colunas e display gráfico de 256 x 192 pixels (elementos gráficos); permite superposição de caracteres (OVER); apresenta recursos de claridade (BRIGHT) e imagem piscante (FLASH)
- Unidade de fita: os programas e dados podem ser armazenados em fita cassete comum, através de gravador convencional. Velocidade de gravação e leitura: 1200 Bauds. Permite gravação de memória, imagens de vídeo, matrizes numéricas e alfanuméricas. Possibilita a integração de dois ou mais programas BASIC através do comando MERGE.
- Cores: 8 cores disponíveis: preto, azul, vermelho, lilás, verde, ciano, amarelo e branco.
- Som: sintetizador de sons controlado por software; comando BASIC que possibilita reproduzir 10 oitavas musicais.
- Joystick: interface incorporada para joystick de 4 posições de movimen-

to e uma de disparo, controláveis também pelo teclado.

Novos Comandos

O BASIC do TK90X possui as mesmas funções do TK82-C e TK85, e ainda dispõe de novos comandos, alguns sendo:

- ATTR - retrata a situação de uma posição especificada na tela
- BIN - converte um número binário para decimal
- BORDER - define a cor da borda da tela
- BRIGHT - intensifica o brilho da cor do fundo da tela
- CIRCLE - traça um círculo na tela
- DATA - permite definir dados a serem lidos por READ
- DEF FN - define funções criadas pelo usuário
- DRAW - traça um arco na tela
- FLASH - gera inversão intermitente de cores
- FN - usado para acessar uma função predefinida por DEF FN
- INK - define a cor dos caracteres na tela
- INVERSE - gera inversão da tela
- MERGE - une um programa na memória com outro em cassete
- OVER - sobrepõe displays da tela
- PAPER - define a cor de fundo da tela
- SOUND - gera notas musicais

IMPRESSÃO COM NANQUIM

• Para se registrar, preto no branco, as informações pelo computador, está sendo cada vez mais utilizada a impressora a nanquim, cuja operação silenciosa é apreciada de modo especial nos escritórios. Em todo o mundo são empregadas cerca de 150.000 impressoras a nanquim, somente da firma Siemens, e novos modelos aparecem constantemente. O aparelho mais novo escreve até 680 caracteres por segundo com tipos de esboço, sendo assim três vezes mais rápido do que a impressoras a nanquim existentes até agora. Mas também com tipos de impressão a máquina chega a 200 caracteres por segundo, o que significa que uma folha normal de papel-ofício é impressa em cerca de quinze segundos. As letras e símbolos surgem através de gotas finíssimas de nanquim, através de 32 injetores, possibilitando uma escrita especialmente clara e representações gráficas exatas.

Pela primeira vez na MICROBITS, um artigo encomendado por um leitor. Esta matéria responde ao pedido de JOÃO BATISTA NEGRO — veja na Seção de Cartas. Que seja o primeiro de uma série. Escrevam-nos!

LINHA VERTICAL EM LINGUAGEM



DE MÁQUINA

A fim de escrever uma linha vertical em linguagem de máquina, segue duas versões, listadas em Assembler. A 1ª fará uma linha vertical com um carácter determinado e a 2ª, mais sofisticada, qualquer frase de 22 caracteres. Chamá-las-emos **VCAR** e **VLIN**, respectivamente.

26 MICROBITS

A rotina **VCAR** é completamente relocável, podendo ser colocada em qualquer localização da memória: acima da **RAMTOP** ou numa linha **REM**.

VCAR — ASSEMBLY

```
LD HL, (16396) ; DFILE
INC HL
LD DE, (16507) ; N° da coluna vertical
LD D, 0 ; Zerar MSB
LD B, 22 ; N° carac. a serem impressos

L1:
ADD HL, DE ; Soma DFILE a N° coluna
LD A, (16508) ; Carácter a ser impresso
LD (HL), A ; Transfere carácter p/ vídeo
LD DE, 33 ; p/ pular 33 caracteres no vídeo

DJNZ L1 ; Decrementa B e salta se não zero
RET ; Retorna ao BASIC
```

VCAR — HEXADECIMAL

2A	0C	40	23	ED	5B	7B	40
16	00	06	16	19	3A	7C	40
77	11	21	00	10	F6	C9	

Conforme a listagem Assembly, note que precisaremos passar os seguintes parâmetros à **VCAR** — Faremos um **POKE** em 16507 a fim de indicar qual a coluna desejada para a impressão e, em 16508, colocaremos o código do carácter a ser impresso na vertical.

Para isso, digite a rotina BASIC:

```
9010 POKE 16507,X
9020 POKE 16508,Y
9030 RAND USR Z
9040 RETURN
```

Esta rotina poderá ser usada em qualquer programa BASIC, onde X será a coluna (entre 0 e 31), Y o código do carácter que encherá a coluna e Z a posição de memória em que está a rotina em linguagem de máquina (ex. 16514).

A rotina **VLIN** que se segue foi planejada para residir na linha 1 **REM** com, pelo menos, 50 caracteres: 28 para a rotina e 22 para os dados da frase que será impressa na vertical.

Procedimentos:

1. Digite, na linha 1 **REM**, 50 caracteres.
2. Inicie a montagem em 16514 e coloque os códigos seguintes em hexadecimal:

2A	7B	40	5D	16	00	2A	0C
40	23	19	11	9E	40	06	16
1A	77	D5	11	21	00	19	D1
13	10	F5	C9				

Se você possui um montador (ou para efeito de estudos), segue-se a listagem Assembly:

```
LD HL, (16507) ; LSB de coluna
LD E, L ; Zerar MSB e
LD D, 0 ; transfere para DE
```


LD HL, (16396) ; DFILE
 INC HL
 ADD HL, DE ; Procura a coluna
 LD DE, L2 ; Início da string
 LD B, 22 ; Contador de caracteres
 a imprimir

L1:

LD A, (DE) ; Carácter da string
 LD (HL), A ; Carácter no vídeo
 PUSH DE ; Preserva endereço vídeo
 LD DE, 33 ; Pula para a
 ADD HL, DE ; próxima linha
 POP DE ; Recupera tabela de
 caracteres

INC DE ; Incrementa-a
 DJNZ L1 ; Decr. B e volta a L1 se
 não = 0

RET ; Retorna ao BASIC

L2:

NOP ; Início da tabela de
 caracteres

3. Digite a rotina BASIC (a qual poderá ser usada em qualquer programa):

```
9000 POKE 16507, COL
9010 DIM A$(22)
9020 INPU A$
9030 FOR F=1 TO 22
9040 POKE 16541+F, CODE A$(F)
9050 NEXT F
9060 RAND USR 16514
9070 RETURN
```

* OBS.: COL = N° da coluna para **VLIN**, entre 0 e 31.

4. Use-a como sub-rotina em qualquer programa BASIC.

A solução acima é apenas *uma* das possíveis. Existem mais rápidas, mais econômicas e mais indexadas. Não obstante, visando atender a maioria, esta nos pareceu a mais didática.

Valter Pereira Vianna trabalha como Analista de Sistemas para PROCONTROL Engenharia de Sistemas S.A. e desenvolve programas na área científica e em linguagem de máquina no seu CP-200 e TK90X.

VALTER PEREIRA VIANNA



LIVRARIA TRIÂNGULO

LIVROS E REVISTAS

com seus bons livros, está de volta.

Fechada há alguns meses pelos seus antigos proprietários, a **Livraria Triângulo** acaba de ser reaberta depois de ser adquirida por 4 jovens que acreditam na cultura nacional e no desenvolvimento tecnológico:

Há mais de 40 anos no mercado, essa livraria iniciou suas atividades no Largo do Tesouro, indo depois para a Galeria Califórnia, onde está até hoje.

A **Triângulo** é uma das livrarias mais tradicionais do País e sempre procurou manter uma linha de alto nível na venda de publicações, sendo visitada por intelectuais e cientistas, que agora podem voltar ao antigo ponto de encontro, selecionando obras nacionais e importadas.

Muito respeitada na área técnica, a livraria possui, no 2º andar, uma coleção de catálogos estrangeiros para os clientes consultarem e encomendarem o que desejam. O pagamento é facilitado em 3 vezes sem juros e funciona todos os dias das 9 às 19 horas e aos sábados das 8h30 às 13 horas. Utiliza também o sistema de entregas a domicílio em São Paulo e também poderão ser encomendados pelo reembolso postal livros nas seguintes áreas:

51 Administração	59 Bioquímica	67 Economia	75 Finanças	83 Medicina
52 Aeronáutica	60 Biblioteconomia	68 Eletrônica	76 Física	84 Metalúrgica
53 Agricultura	61 Botânica	69 Engenharia Civil	77 Geociências	85 Química Geral
54 Alimentos	62 Ciências Sociais	70 Engenharia Elétrica	78 Hidráulicas	86 Química Industrial
55 Astrologia	63 Computação	71 Engenharia Mecânica	79 Hobby	87 Termotécnica
56 Arte	64 Cozinha	72 Esportes	80 Literatura	88 Veterinária
57 Automóveis	65 Dicionários	73 Estatística	81 Marketing	89 Zoologia
58 Biologia	66 Ecologia	74 Farmácia	82 Matemática	

Para receber catálogos e informativos sobre obras de seu interesse, sublinhe as áreas acima, e envie fotocópia preenchida à máquina ou letra de forma para Caixa Postal nº 30.317/CEP 01051 São Paulo, SP.

Livraria Triângulo

R. Barão de Itapetininga, 255
 Lojas 23/24 - Galeria Califórnia
 Cep 01042 São Paulo: SP
 Tel. (011) 231-0922

Nome															
End.															
						Bairro									
Cep				Mun.				Estado							
Firma						Divisão									

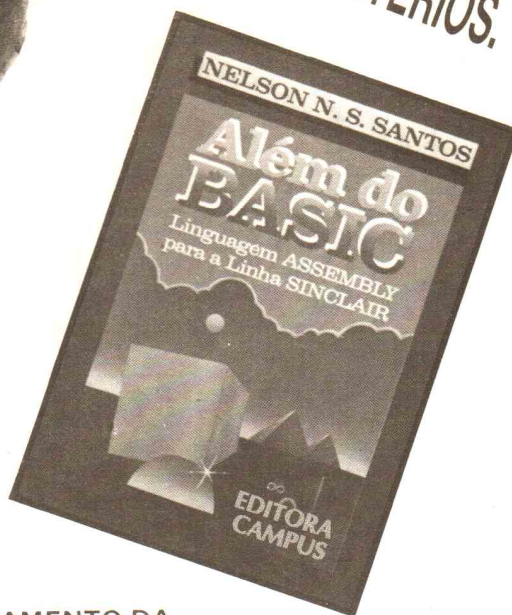
fotos

A linguagem do presente

Educação/cultura

À já extensa bibliografia da área de informática acaba de ser acrescentado mais um título: *Além do Basic - Linguagem Assembly para a Linha Sinclair*, escrito por Nelson do Nascimento Silva dos Santos, um professor de química e matemática apaixonado por computação. Há algum tempo, Nelson é responsável pela coluna *Além do Basic* na revista *Microbits*, e agora lança este livro que, apesar de exigir uma certa iniciação no assunto, tem uma vantagem sobre os demais: é didático, o que facilita muito o seu entendimento. Através de exercícios em ordem crescente de dificuldade, o leitor tira suas dúvidas sobre a linguagem Assembly, ao mesmo tempo em que se familiariza com a informática, a ciência que já é do presente.

CONHEÇA
O PRIMEIRO LIVRO
QUE REALMENTE ENSINA
LINGUAGEM DE MÁQUINA,
SEM MISTÉRIOS.



MAIS UM LANÇAMENTO DA

Editora Campus Ltda.

MICROBITS

UMA BOA NOTÍCIA PARA VOCÊ:
Os leitores que assinam MICROBITS
economizam tempo e dinheiro!

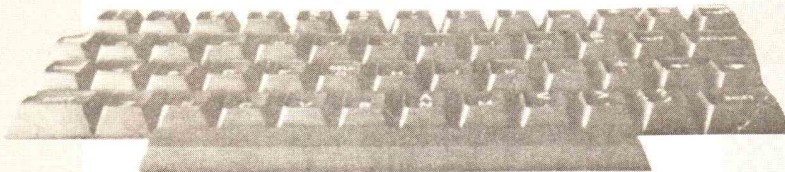
- 1) LEMBRE-SE: Você irá receber cada número comodamente em sua casa, tão logo seja publicado.
- 2) COMPARE: Preço Assinatura (6 números) Cr\$ 52.000
Preço Nº Avulso Cr\$ 9.600
- 3) SOLICITE: Ainda hoje a sua assinatura.

MICROBITS

PEDIDOS DE ASSINATURA PARA:
Editora Campus Ltda.

Rua Barão de Itapagipe, 55 - Rio Comprido - 20261 Rio de Janeiro - RJ - Tel.: (021) 284 8443

Classificados



COMPRA, VENDA, TROCA, OFERTAS,
PRESTAÇÃO DE SERVIÇOS.
Classifique como quiser, mas use este espaço, que é seu.

SOFTWARE

1

- **METASOFT** – Vendemos fitas para computador. Temos "Demon of Vega", Pirata, Pac-Man e muitas outras. Telefones para contato: (011) 571-0003, 275-0378.
- **VENDO** e troco programas e jogos para TK82, 83, 85, CP-200 e compatíveis; TK 2000, TRS-80, APPLE e compatíveis. Escreva solicitando catálogo explicativo. Sérgio Akira Tomida, Rua Prudente de Moraes, 65, Bairro Jardim D'Abril, 06000 – Osasco – SP.
- **SOFT** para TK e compatíveis. Mais de 80 títulos. Maiores informações pelo telefone (0192) 75-2205 com Charles ou (0192) 75-3698 com Sérgio, na parte da tarde.
- **VENDO** ou troco programas para linha Sinclair. Tenho cerca de 500 programas, entre jogos, aplicativos, educacionais, etc. Pode ser em fita ou em listagem de impressora Timex. Juscelino Kilian, Caixa Postal 409, 96500 – Cachoeira do Sul – RS.
- **ZX-SPECTRUM**. Procure-se programas para o ZX-Spectrum. Caso você tenha programas para este computador, entre em contato comigo através do seguinte endereço: Alexandre Gromow, Caixa Postal 18507, São Paulo – SP.
- **TROCO** e vendo programas para CP-200, TK e similares. Possui programas aplicativos e de lazer. Maurício Ferreira Leite, Rua Jaú, 230, Santa Efigênia, 30000 – Belo Horizonte – MG.
- **VENDO** software para Sinclair. Programas inéditos, 10 a escolher (entre mais de 300) em fita, por menos de 1 ORTN. Também troco jogos em Alta Resolução para micros adaptados. Peça catálogo *grátis* à A. M. Brito, SHIN QI-13 – cj. 04/cs. 04, 71500 – Brasília – DF.
- **TROCO** programas de lógica Sinclair e vendo de lógica TRS-80. Gostaria também de trocar idéias e informações. Marlos Medeiros, Rua Frei Orlando, 599, 60000 – Fortaleza – CE.

HARDWARE

2



- **ADAPTAMOS** vídeo inverso, controle automático de gravador, LED indicador de funcionamento, botão RESET. Os interessados devem entrar em contato com: Erinaldo Alencar Rodrigues, Rua General Sampaio, 669, Centro, 60000 – Fortaleza – CE.

Dicas

ENVIEM COLABORAÇÕES!

Editar linhas é tarefa das mais desagradáveis para o programador, e não se pode escrever um programa para ajudar neste trabalho, porque EDIT não é acessível como linha de programa. Certo? **ERRADO!** Um pequeno truque, usando os endereços da ROM, torna isto possível.

Se você tem muitas linhas a consertar na depuração de um programa, digite (naturalmente 9000 é apenas uma sugestão):

```
9000 PRINT "QUAL LINHA EDITAR?_";
9010 INPUT N
9020 PRINT N
9030 POKE 16394,N-256*INT(N/256)
9040 POKE 16395,INT(N/256)
9050 RAND USR 1476
```

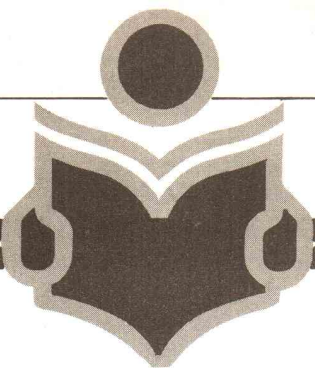
As linhas 9030 e 9040 tratam a variável do sistema E.PPC, que contém o número da linha onde está o cursor – consulte o manual.

A linha 9050 é equivalente a EDIT. Verifique no mapa da ROM que 1476 (05C4) é o endereço da principal rotina de edição.

Vamos agora repetir a listagem hexa da dica Quantos bytes ainda restam?", da **MICROBITS ANO II, Nº 3**, que saiu com um errinho. Ah, Murphy...

```
16514 2A 14 40 E5 C1 2A 04 40
16522 A7 ED 42 E5 C1 C9
```

Nelson N. S. Santos



• DICIONÁRIO DO BASIC SINCLAIR

Autor: LIMA, Délio Santos
Editora: Micron Eletrônica

Este livro é justamente como descrito pelo seu título — um dicionário dos comandos ou instruções disponíveis nos micros compatíveis com o SINCLAIR ZX 81.

Os assuntos iniciais tratados são:

- O que é BASIC SINCLAIR
- Compatibilidade (ou não!) de software entre o ZX81 e suas versões nacionais: — os TK's, CP-200, NE-Z8000, Ringo e AS-1000.
- A divisão do BASIC SINCLAIR em "keywords", caracteres, instruções etc.
- As diferenças entre instruções e comandos.
- As funções (matemáticas etc.).
- Os operadores (aritméticos, relacionais e lógicos).
- Os caracteres.

Em seguida há o dicionário propriamente dito. As "keywords" (palavras-chave) são listadas alfabeticamente, com descrição da função, instrução ou operador e exemplo do seu uso, normalmente com um pequeno programa ilustrativo.

Os operadores lógicos (AND, OR, NOT), raramente utilizados além das instruções mais simples como:

```
10 IF A = 3 AND B = 2 THEN ...
```

são detalhados. Por exemplo, as duas linhas

```
10 LET X = 1
```

```
20 IF Y = 0 THEN LET X = 0
```

podem ser substituídas por uma única linha:

```
10 LET X = 1 AND Y
```

Um segundo exemplo do uso seria

```
10 LET X$ = Y$ AND A
```

Neste caso, se o segundo operando (A) for zero, a expressão assume o valor 0 (ou o mesmo que X\$=""). Ao

contrário, se não for zero então X\$ será igual a Y\$.

No final do livro há seções sobre

- Convertendo outros BASICs (não há READ, DATA, RESTORE no Sinclair; LEFT\$, RIGHT\$ e MID\$ têm que ser convertidos para X\$ (A TO B), por exemplo; etc.).
- Contando os bytes.
- Economizando memória.

Embora mais indicado para usuário principiante ou aquele que quer converter programas de um outro tipo de BASIC, o conteúdo deste livro pode ser útil mesmo para quem já conhece bem o seu micro. Confesso que às vezes, confundo-me entre CODE e CHR\$. E o erro na ROM quando LPRINT é usado com números fracionários menores que 0,01, pode surpreender qualquer usuário tirando uma lista de dados na sua impressora. O dicionário, nestes e em outros casos, dá a informação necessária para se tirar dúvidas.

David Thomas Anderson



• ALÉM DO BASIC — Linguagem Assembly para a Linha Sinclair

Autor: SANTOS, Nelson N. S.
Editora: Campus

Para quem já necessita informações "ALÉM DO BASIC", Nelson N. S. Santos editou, pela Campus, um livro que mostra como você pode "conversar" com seu micro.

Composto de 9 capítulos, ele mostra gradativamente, através de comparações com BASIC, como utilizar a linguagem de máquina para melhorar seus programas.

Comentando em cada capítulo as funções existentes, e com uma bateria de exercícios que faz com que o programador saiba na íntegra o que está acontecendo com o seu micro, o livro proporciona ao leitor todos os truques necessários para se programar em linguagem de máquina.

Além de 9 capítulos que dão "dicas" excelentes, tanto para o iniciante co-

mo para o programador mais experiente, ele apresenta em 16 apêndices, entre outros os assuntos: tabela de conversão decimal/hexadecimal, tabela de complemento de dois, FLAG's afetados pelas instruções e em que bit ocorreu a modificação, conjunto de instruções do Z-80 em ordem alfabética e numérica, etc. Fornece assim um ótimo material de consulta, que se torna indispensável ao programador.

Sérgio Augusto Moraes



• 40 ROTINAS EM LINGUAGEM DE MÁQUINA

Autor: CARVALHO, Douglas de
Editora: Micron Eletrônica

O livro se propõe a mostrar subrotinas para auxiliar o desenvolvimento de programas para micros de lógica SINCLAIR. Tem-se aqui rotinas interessantes que implementam tanto seus programas em BASIC quanto em linguagem de máquina.

Além disso, consta de uma introdução elucidativa quanto ao uso de registradores, microprocessadores, instruções e modos de endereçamento, dando uma visão muito boa dos recursos que a linguagem de máquina pode oferecer.

Para cada rotina o autor descreve seu funcionamento, condições de entrada e saída, listagem em Assembly e em hexadecimal, podendo, assim, utilizar imediatamente recursos como: SCROLL e rolagem de tela em todas as direções; "checksum" de áreas de memória; multiplicação e divisão de números de 16 bits (múltipla precisão), e outras mais.

Enfim, os iniciantes terão chance de usar os programas sem conhecimento prévio de linguagem de máquina, podendo vir a interessar-se por essa nova opção. Os mais experientes terão oportunidade de estudar a confecção de rotinas realmente úteis para o programador que deseja aprofundar-se em Assembly.

Valter Pereira Vianna

EDIÇÕES AVULSAS

Complete a sua coleção de MICROBITS solicitando os números que lhe faltam.
Verifique a matéria publicada nas edições anteriores:

Ano I – Nº 1

- Introdução ao CLUBE NACIONAL DOS TK/NE/SINCLAIR
- Programa "PLOTAR"

Ano I – Nº 2

- Linguagem de máquina: Onde e Como?
- "Benchmark" comparada a linguagem de máquina
- Como Criar um Cadastro (Parte 1)
- Programa "CORRIDA DE CARROS"

Ano I – Nº 3

- Programa "CONTAS A PAGAR"
- Como Proteger Seus Dados
- Conserto de Teclado Defeituoso
- Teclado Musical
- Conversão Hexadecimal-Decimal
- Medidor de Intensidade de Saída do Gravador para "LOAD"

Ano I – Nº 4

- Armazenar a Imagem Numa Variável
- O TK85 da Microdigital – Vale a Pena?
- Testando o TIG-LOADER
- Programa "SENHA"
- Programa "VALOR PRESENTE"

Ano I – Nº 5

- BENCHTEST: ANÁLISE DO RINGO
- Maior Resolução Gráfica
- Alta Resolução no Vídeo
- VU Incrementado
- Codificador de Texto
- Submarino Nuclear
- Curvas Paramétricas

Ano I – Nº 6

- Como Criar Caracteres Especiais no Seu Micro – Parte I
- Linguagem de Máquina: Como Usar a Tela
- Benchtest: CMS/ZX da Polimicro
- Como Criar um Cadastro (Final)
- Sistema de Equações

IMPORTANTE

Qualquer dos seis números publicados no Ano I pode ser obtido mediante envio de cheque nominal diretamente a:

MICRO BITS INFORMÁTICA LTDA.
CAIXA POSTAL 12.464
CEP 04798 – SÃO PAULO – SP

Ano II – Nº 1

- Como Criar Caracteres Especiais no Seu Micro – Parte II
- Salvando e Restaurando Telas
- Benchtest: Teclado da Speed
- Consumo
- Crie uma Linha REM
- Controle de Gravador K7
- Benchtest: Gluckstick Digital
- PERT
- Jogo de Inversão
- Caracteres de Vídeo
- Desenhe na Tela
- Interface para Telegrafica (C.W.)
- Números Primos

Ano II – Nº 2

- A Grande Arma
- Como Criar Caracteres Especiais no Seu Micro – Parte III
- Teclados de Membrana
- FORCA em Linguagem de Máquina
- Além do BASIC
- Programa PIRATAS
- Avenida II

Ano II – Nº 3

- Desafio
- A Linha REM
- Torres de Hanói
- Arquivo de Custos
- TK na Abertura da Fórmula 1
- Programa Integral
- Além do BASIC
- TRON – Uma Fantástica Aventura
- Aproximação de Romberg
- Programa Ordenador

IMPORTANTE

Os números publicados a partir do Ano II nº 1 podem ser obtidos mediante envio de cheque nominal diretamente a:

Editora Campus Ltda.
Rua Barão de Itapagipe, 55
CEP 20261 – Rio de Janeiro – RJ
Tel.: (021) 284 8443

- Números anteriores serão vendidos ao preço corrente da revista.
- Este valor já inclui as despesas de correio.

 **Editora Campus Ltda.**

PONHA OS MICROS NO SEU DEVIDO LUGAR



Os livros da **CAMPUS** fazem o seu micro trabalhar.

MARQUE OS LIVROS QUE DESEJA RECEBER:

Linha Apple:

- Coletta, P. — Jogos Gráficos para o Apple . . . Cr\$ 49.000
- Cook, R. e Hartnell, T. — Como Programar o seu Apple . . . Cr\$ 25.500
- Kantaris, N. — Manual Apple em BASIC . . . Cr\$ 32.500

Linha Sinclair:

- Hughes, C. — Primeiros Passos com o seu SINCLAIR (TKs) . . . Cr\$ 30.500
- Christmann, R. U. — BASIC SINCLAIR . . . Cr\$ 29.500
- Meili, E. A. — APLICALC, um Software Educacional, Pessoal e Profissional em BASIC . . . Cr\$ 33.500
- Karsten, L. — Programas Administrativos em BASIC SINCLAIR . . . Cr\$ 36.000
- Loiola, C. R. A. — Rotinas Matemáticas em BASIC para Micros . . . Cr\$ 62.000
- Botelho, A. J. L. — Desafio: Os Mais Excitantes Jogos em BASIC . . . Cr\$ 54.000
- Santos, N. N. S. — Além do BASIC — Linguagem ASSEMBLY para a Linha SINCLAIR . . . Cr\$ 34.000

Linha TRS

- Alexander, D. C. — Programação em ASSEMBLER e Linguagem de Máquina (2ª Edição) . . . Cr\$ 42.500
- Chance, D. — 30 Programas em BASIC para Computadores Pessoais . . . Cr\$ 57.500
- Gratzler, E. A. e Gratzler, T. G. — BASIC Rápido: Além do BASIC TRS-80 . . . Cr\$ 51.500
- Sawusch, M. — 1001 Aplicações para o seu Computador Pessoal . . . Cr\$ 49.000

ASSINALE A SUA OPÇÃO DE COMPRA:

- Mediante envio de cheque nominal à **EDITORA CAMPUS LTDA**, (o porte é por nossa conta e sua encomenda chega mais rápido)
Cheque nº Banco
- Mediante pagamento contra recebimento dos livros pelo reembolso postal, acrescido das despesas postais.

PREENCHA AQUI:

Nome:

CEP: CIDADE:

ESTADO:

IMPORTANTE: Preços válidos por tempo limitado.

Editora Campus

Rua Barão de Itapagipe, 55 - 20261
Rio de Janeiro - RJ - Tel.: (021) 284-8443

ESTE ANÚNCIO PODE SER XEROCADO